REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
18-01-2011	Final Report	12-Jul-2006 - 11-Sep-2010

4. TITLE AND SUBTITLE
Final Report - Improving Environmental Model Calibration and Prediction

5a. CONTRACT NUMBER
W911NF-06-1-0306

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Jeffrey S. Baggett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Wisconsin - La Crosse
Office of Sponsored Programs
1725 State St.
La Crosse, WI 54601 -

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
51068-EV.3

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
First, we have continued to develop tools for efficient global optimization of environmental models. Our algorithms are hybrid algorithms that combine evolutionary strategies for global search with derivative-free methods for local search in novel and efficient ways. Results thus far are promising and are pointing the way toward practical hybrid optimization tools for environmental models.

15. SUBJECT TERMS
optimization, environmental model calbration, evolutionary strategy, Bayesian optimization

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jeff Baggett
UU	UU	UU	UU		19b. TELEPHONE NUMBER: 608-785-8393

Report Title

Final Report - Improving Environmental Model Calibration and Prediction

ABSTRACT

First, we have continued to develop tools for efficient global optimization of environmental models. Our algorithms are hybrid algorithms that combine evolutionary strategies for global search with derivative-free methods for local search in novel and efficient ways. Results thus far are promising and are pointing the way toward practical hybrid optimization tools for environmental models.

Second, we are applying function approximation techniques to improve the efficiency of Monte Carlo Markov Chain simulations used in Bayesian optimization of environmental models. Very simple preliminary experiments have yielded 20-30% reduction in computational effort for well converged estimates of model parameter probability distributions.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Number of Papers published in peer-reviewed journals: 0.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Hybrid Optimization using Evolutionary and Approximate Gradient Search for Expensive Functions; Jeff S. Baggett and Brian E. Skahill, Second International Conference on Engineering Optimization, Lisbon, Portugal; September, 2010.

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 1

(d) Manuscripts

Hybrid Optimization using an Evolutionary Strategy and Surrogate Assisted Local Search; Jeff S. Baggett and Brian E. Skahill, submitted to Engineering Optimization; September, 2010.

Number of Manuscripts: 1.00

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Post Doctorates

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Final Report - Improving Environmental Model Calibration and Prediction

Jeffrey S. Baggett

Contents

1 Statement of Problem

First, in the area of traditional methods model calibration we have continued developing tools for efficient global optimization of complex environmental models. In particular, we are refining hybrid optimization algorithms that blend evolutionary strategies for global optimization with local search based on a locally constructed surrogate model. A conference presentation was given on this topic and an article appeared in the conference proceedings ([2]; attached in Appendix A). Moreover, a journal article was prepared and submitted, ([1]; attached in Appendix B). Most recently we have begun utilizing state of the art techniques for constructing certain local quadratic polynomial models with special approximation properties [14] to utilize in a local search step that is interwoven between iterations of an evolutionary strategy.

Second, we are exploring surrogate modelling techniques for the acceleration of Bayesian model calibration. In Bayesian model calibration Monte Carlo simulation is used to infer approximate posterior probability distributions of the parameters given the observed data, but many model runs are required for this inference. We have recently begun collecting data from previous model runs to interpolate future estimates of the posterior probability. These estimates are used to filter out unnecessary model runs, thus increasing the efficiency of the Monte Carlo simulation.

2 Results Summary

2.1 Introduction

Computer-based calibration of environmental models generally involves minimization of an "objective function"-a measure of model-to-measurement misfit. In the watershed modeling context it could be specified as differences between measured and modeled stream flows (see, for example, [3, 5]).

An important consideration in assessing the performance of parameter estimation software is that of run time. Parameter estimation software must run the model to be calibrated many times in the course of minimizing the objective function. For models in which the objective function surface contains multiple local minima, model run times are high, or when multiple prediction specific calibrations must be conducted [8], it is particularly important that the parameter estimation software conducts a minimal number of model runs.

Our parameter estimation software is called the Model Independent Calibration and Uncertainty analysis Toolbox (MICUT) and it is compatible with the PEST model independent protocol [4] making it immediately usable by a large community. MICUT, while compatible with PEST, is distinctive in that it employs more efficient algorithms for calibration of complex models than PEST [9]. If a single model run takes hours or days to execute this is significant. The primary focus of my research is on the further develpment of *efficient* tools for model calibration and uncertainty analysis.

For calibration of models in which the objective function is not smooth or has many local minima the use of traditional gradient-based search, even when combined with a global, stochastic search strategy is at best inefficient and at worst fails completely. For such models, one approach is to use evolutionary strategies, in which survival of the fittest is used to adapt the best parameter sets until a global minimizer is found. One of the best evolutionary strategies to date is the Covariance Matrix Adaptation Evolution Strategy (CMAES; [6]).

2.2 The Covariance Matrix Adaptation Evolution Strategy

The CMAES is an evolution strategy that adapts the covariance matrix of a normal search distribution. The user specifies a population size of λ individuals. The randomly selected initial population is $x_{1:\lambda}^{(0)}$. After evaluating the objective function, the best μ individuals (with smallest values) are selected and their center of mass is computed by $\langle x \rangle_W^{(0)} = \sum_{k=1}^{\mu} \omega_k x_{k:\lambda}^{(0)}$. The recombination weights, ω_k, are positive and sum to one. After selection and recombination, a new population is created accordig to $x_{1:\lambda}^{(t+1)} = \langle x \rangle_W + \sigma^{(t)} \sqrt{C^{(t)}} z_{1:\lambda}^{(t)}$. The covariance matrix, $C^{(t)}$, and the step size, $\sigma^{(t)}$ are updated after each generation [6]. The $z^{(t)}$ are randomly sampled from the multivariate standard normal distribution.

On convex-quadratic functions CMAES converges log-linearly after an adaption time that scales between 0 and n^2, where n is the number of parameters. For multimodal objective functions, CMAES does a surprisingly good job at finding the global minimum, but the population size must be increased significantly and with considerable problem variation, to ensure global convergence [7]. For the synthetic problems and watershed models we have thus far considered the CMAES algorithm yields slower convergence on average than the hybrid algorithm using the Levenberg-Marquardt for local search and the Multi-Level Single Linkage algorithm for global search [9], thus we are attempting to accelerate the convergence of CMAES while maintaining its robustness for finding good approximations to the global minimizer.

2.3 Hybrid Evolutionary Strategy and Local Search

Optimization of expensive functions is more feasible with algorithms that require fewer evaluations of the objective function. In that spirit, we propose a hybrid evolution strategy. As in [10, 11, 16], the proposed algorithm combines an evolution strategy (ES) with a gradient search technique. The proposed algorithm differs from the algorithm of Tahk, et al [10], in that it uses a local radial basis function approximation to the objective function, instead of finite differences, to compute approximate first and second derivatives to the objective function surface. The derivative information is used to propagate a gradient individual alongside the evolving population. The gradient individual is included for possible selection each generation. The use of radial basis functions allows the derivatives to be estimated without imposing any special structure on the evolving population of points as is required in [10]. This enables the use of the CMAES which proved to be rather sensitively tuned and evidently breaks when the evolving population is perturbed strongly from its random Gaussian distribution.

Tests on a small suite of standard test functions and a hydrologic application show that this hybrid approach can greatly accelerate the covariance matrix adaptation evolution strategy (CMAES) requiring many fewer evaluations of the objective function than standard CMAES at a greater cost per optimization iteration than standard CMAES. However, the extra computational time would be incidental for a complex environmental model.

Complete details of the method and the numerical tests can be found in the attached conference paper [2] in Appendix A and manuscript submitted to Engineering Optimization [1] in Appendix B.

The current version of the hybrid CMAES/rbf-assisted local search requires that the objective function be reasonably smooth, but this is often not the case for environmental models. Two new variations of the hybrid algorithm are currently being developed that utilize local models requiring less smoothness [14, 15]. Preliminary efforts using these local models to calibrate real surface water

hydrological models have been very promising. Future work will include incorporating one or both of these new hybrid algorithms into our Model Independent Calibration and Uncertainty analysis Toolbox where they will available for application to problems of interest for the Corps of Engineers and other researchers.

2.4 Efficiency enhancements for Bayesian Model Calibration

In Bayesian Model Calibration, the model parameters are treated as random variables with unknown distributions. If any information about the parameters is known in advance this is specified as a prior distribution. In the hydrologic context usually only bounds on the parameters are known so only a uniform or uninformative prior distribution is typically assumed for each parameter. The idea, then, is to find the posterior probability distribution of the parameter sets given the observed data. Because the probability distribution of the data is unknown this is often done by Monte Carlo simulation using a so-called Monte Carlo Markov Chain (MCMC) sampler. There are many such samplers, of which one that is particularly popular in the hydrological community is the Shuffled Complex Evolution Metropolis (SCEM-UA) sampler [13]. As with all MCMC samplers, the convergence to the posterior distribution of the parameters can be quite slow and may require thousands or tens of thousands of model evaluations. Thus, while a complete characterization of the probability distribution is desirable, it may be prohibitively expensive.

Fortunately, many of the newly sampled points in the MCMC sequence may be from areas in parameter space in which the posterior distribution is already well-sampled and additional sampling may not lead to new information. To check for this, we store all previously sampled points in a database along with the values of the log-likelihood function (used in the estimation of the posterior probability). At each newly sampled point locally weighted projection regression (LWPR; [12]) is used to estimate the posterior probability. Parameter points which are unlikely to be accepted in the current Markov chain based on their LWPR estimated values are rejected. The actual model is evaluated and the real posterior probability calculated for parameter points which were judged possibly acceptable by their LWPR estimates. To assure convergence, of the Markov chains it was found necessary to, at random, occasionally evaluate the true posterior probability at a proportion of the rejected points as well. This approach resulted in a 20%-30% savings of model evaluations while converging to comparable posterior probability distributions. While this was a good first effort, we are attempting to refine the approach to yield at least 50% savings before publishing the method. A preliminary report is attached in appendix C.

References

[1] J.S. Baggett and B.E. Skahill. Hybrid optimization using an evolutionary strategy and surrogate assisted local search. *Submitted to J. Engineering Optimization*, 2010.

[2] J.S. Baggett and B.E. Skahill. Hybrid optimization using evolutionary and approximate gradient search for expensive functions. In *Proc. 2nd Int. Conf. Eng. Opt.*, 2010.

[3] D.P. Boyle, H.V. Gupta, and S. Sorooshian. Toward improved calibration of hydrologic models: combining the strengths of manual and automatic methods. *Water Resour. Res.*, 36(12):3663–3674, 2000.

[4] J. Doherty. *Model Independent Parameter Estimation. Fifth edition of user manual.* Watermark Numerical Computing.

[5] J. Doherty and J.M. Johnston. Methodologies for calibration and predictive analysis of a watershed model. *J. American Water Resources Association*, 39(2):251–265, 2003.

[6] N. Hansen. The cma evolution strategy: A tutorial. Available from: `http://www.lri.fr/~hansen/cmatutorial.pdf`, 2010.

[7] N. Hansen and S. Kern. Evaluating the cma strategy on multimodal test functions. In X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J Merelo-Guervos, J.A. Bullinaria, J. Rowe, P. Tino, A. Kaban, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 282–291. Springer Verlag, 2004.

[8] C. Moore and J. Doherty. The cost of uniqueness in groundwater model calibration. *Adv. Water Resour.*, 29(4):605–623, 2006.

[9] B.E. Skahill, J.S. Baggett, S. Frankenstein, and C.W. Downer. More efficient pest compatible model independent model calibration. *Environmental Modelling and Software*, 24(4):517–529, 2009.

[10] M.-J. Tahk, M.S. Park, H.W. Woo, and H.J. Kim. Hessian approximation algorithms for hybrid optimization methods. *Engineering Optimization*, 41(7):609–633, 2009.

[11] M.-J. Tahk, H.W. Woo, and M.S. Park. A hybrid optimization method of evolutionary and gradient search. *Engineering Optimization*, 39(1):87–104, 2007.

[12] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634, 2005.

[13] J.A. Vrugt, H.V. Gupta, W. Bouten, and S. Sorooshian. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resources Research*, 39, 2003.

[14] Stefan M. Wild. MNH: a derivative-free optimization algorithm using minimal norm hessians. In *Tenth Copper Mountain Conference on Iterative Methods*, April 2008. Available at http://grandmaster.colorado.edu/ copper/2008/SCWinners/Wild.pdf.

[15] Stefan M. Wild, Rommel G. Regis, and Christine A. Shoemaker. ORBIT: optimization by radial basis function interpolation in trust-regions. *SIAM J. on Scientific Computing*, 30(6):3197–3219, 2008.

[16] H.W. Woo, H.H. Kwon, and M.-J. Tahk. A hybrid method of evolutionary algorithms and gradient search. In S.C. Mukhopadhyay and G. Sen Gupta, editors, *Proc. 2nd Int. Conf. Autonomous Robots and Agents*, 2004.

Appendix A

Hybrid Optimization using Evolutionary and Approximate Gradient Search for Expensive Functions.

Jeff S. Baggett and Brian E. Skahill

Second International Conference on Engineering Optimization

Lisbon, Portugal

Paper appears in conference proceedings and was presented September, 2010.

2nd International Conference on Engineering Optimization
September 6-9, 2010, Lisbon, Portugal

Hybrid Optimization using Evolutionary and Approximate Gradient Search for Expensive Functions

Jeff S. Baggett and Brian E. Skahill

University of Wisconsin - La Crosse, La Crosse, WI, USA, baggett.jeff@uwlax.edu
US Army Engineer Research and Development Center, Vicksburg, MS, USA, Brian.E.Skahill@usace.army.mil

Abstract

Optimization of expensive functions is more feasible with algorithms that require fewer evaluations of the objective function. In that spirit, this paper proposes a hybrid evolution strategy. As in [11, 9, 10], the proposed algorithm combines an evolution strategy (ES) with a gradient search technique. The proposed algorithm differs in that it uses a local radial basis function approximation to the objective function to compute approximate first and second derivatives to the objective function surface. The derivative information is used to propagate a gradient individual alongside the evolving population. The gradient individual is included for possible selection each generation. Tests on a small suite of standard test functions and a hydrologic application show that this hybrid approach can greatly accelerate the covariance matrix adaptation evolution strategy (CMAES). This hybrid approach is flexible and requires little modification of an existing evolution strategy; thus, it does not seem to alter negatively affect convergence when an objective function does not have sufficient smoothness for derivatives to yield useful descent information.
Keywords: Hybrid algorithms; Evolution strategies; Quasi-Newton method

1. Introduction

For multi-modal fitness functions evolution strategies are known to be reliable, but slow for approximating global optima. A large population size is required for the strategy to explore the real parameter space, but this slows local convergence. In contrast, classical gradient-based algorithms are exploitative in nature and converge quickly to local minima, but they are not good at finding the global minimum.

The algorithm proposed in this paper follows the hybrid evolution strategy first outlined in [11], but differs in two respects: first and most importantly, a more accurate and more flexible method is utilized to approximate the derivatives required for the local search, and second, the evolution strategy is the covariance matrix adaptation evolution strategy (CMAES) which is known to be particularly effective at approximating global optima. CMAES is especially effective at locating global optima when used in conjunction with a population doubling strategy as described in [1].

The derivative estimation method will be explained first and it will be shown how the approximated derivatives are used to propagate the gradient individual between generation. Next the hybrid algorithm will be summarized. We have implemented the hybrid algorithm in CMAES, and it will be shown how the new hybrid algorithm performs on a suite of test functions in 10 dimensions. Finally, we briefly demonstrate an application in hydrological modeling.

2. Gradient Individual using Radial Basis Functions

The foundation of this optimization algorithm is the usual evolution strategy in which new offspring are produced at each generation by recombination and mutation (see Figure 1). The objective function is evaluated at each of these offspring and the fittest offspring are selected as parents for the next generation. In the hybrid approach, an additional individual, called the gradient individual [9], is propagated by a different mechanism each generation. The gradient individual, x_g^t, is either the fittest offspring of the current generation (individual with lowest function value) or the gradient individual from the previous generation. From information gathered during the evolution of the population the first and second derivatives of the objective function are estimated at x_g^t and used to perform an update of the gradient individual which is hopefully moved closer to a stationary point.

As an evolution strategy proceeds, it typically does not use the previously evaluated points beyond the current generation; however, in our hybrid strategy we store the last N points and their evaluated functions values in a database. To update the gradient individual we use a k-nearest neighbor local

function approximation of the objective function using the k nearest neighbors (Euclidean distance) of x_g^t in the database to construct a cubic radial basis function (RBF) approximation:

$$s(x) = \sum_{i=1}^{k} w_i \phi\left(\|x - x_i\|_2\right) + p(x), \quad x \in \mathbb{R}^n \tag{1}$$

where $x_i, i = 1, 2, \ldots, k$ are the k nearest neighbors of x_g^t in the n-dimensional search space, p is in Π_2^n (the linear space of polynomials in n variables of degree less than or equal to 2), and $\phi(r) = r^3$. Cubic radial basis functions were selected not only for their simplicity and differentiability, but also because they have been used successfully as surrogate models for pre-evaluating function values to lessen the number of function evaluations required by an evolution strategy [8].
Define the matrix $\Phi \in \mathbb{R}^{k \times k}$ by

$$(\Phi)_{ij} := \phi\left(\|x_i - x_j\|_2\right), \quad i, j = 1, \ldots k. \tag{2}$$

Let $\hat{n} = (n+1)(n+2)/2$ be the dimension of Π_2^n, let $p_1, \ldots, p_{\hat{n}}$ be a basis of this linear space, and define the matrix $P \in \mathbb{R}^{k \times \hat{n}}$ as follows:

$$P_{ij} := p_j(x_i), \quad i = 1, \ldots, k; j = 1, \ldots, \hat{n}. \tag{3}$$

In this model, the RBF that interpolates the points $(x_1, f(x_1)), \ldots (x_k, f(x_k))$ is obtained by solving the system

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} w \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0_{\hat{n}} \end{pmatrix} \tag{4}$$

where $F = (f(x_1), \ldots, f(x_k))^T, w = (w_1, \ldots, w_k) \in \mathbb{R}^k$ and $c = (c_1, \ldots, c_{\hat{m}})^T \in \mathbb{R}^{\hat{m}}$. Powell [7] gives sufficient and necessary conditions for the system above to be uniquely solvable, but in practice the real issue can be that the coefficient matrix above becomes ill-conditioned. However, we have found that simply rescaling and shifting the points $x_1, \ldots, x_k$ so that they all lie in $[-1, 1]^n$ is usually sufficient to address this issue.
Once the RBF, $s(x)$, has been determined by Eq.(1), then $s(x)$ is differentiated analytically to determine approximations to the gradient and Hessian of the objective function, $f(x)$. For the gradient vector, g, evaluated at the gradient individual, x_g^t:

$$g_i = \left(\nabla f(x_g^t)\right)_i = \frac{\partial}{\partial x_i} f(x_g^t) \approx \left(\nabla s(x_g^t)\right)_i = \frac{\partial}{\partial x_i} s(x_g^t), \quad i = 1, \ldots, n \tag{5}$$

For the Hessian matrix, H, evaluated at the gradient individual, x_g^t:

$$H_{ij} = \left(H(x_g^t)_{ij}\right) = \frac{\partial^2}{\partial x_i \partial x_j} f(x_g^t) \approx \frac{\partial^2}{\partial x_i \partial x_j} s(x_g^t), \quad i, j = 1, \ldots, n \tag{6}$$

Similar techniques for derivative approximation are routinely used in the solution of partial differential equations using so-called meshless methods. Moreover, such approximations can be spectrally accurate [3].
Once the offspring and their function values from the current generation have been appended to the database, we construct the RBF as above and determine the gradient and Hessian approximated at the current gradient individual x_g^t. Finally, a new gradient individual is found by the standard update:

$$x_g^{t+1} = x_g^t - H^{-1} g. \tag{7}$$

The new gradient individual is then added to the current generation of offspring for possible selection. When the population, and hence the points in the database, are sufficiently close to a minimum then the gradient and Hessian can be accurate and can yield fast local convergence to the minimum. However, when the evolving population is far from a local minimum, then the gradient and Hessian tend to be inaccurate and/or at least the update does not lead to a minimum. It's worth noting that the addition of this gradient individual to the offspring pool for selection seems to never be harmful to the search.
This method of gradient-based search is similar to a quasi-Newton method, but in quasi-Newton methods the first derivatives of the function are usually available (or approximated by centered finite differences),

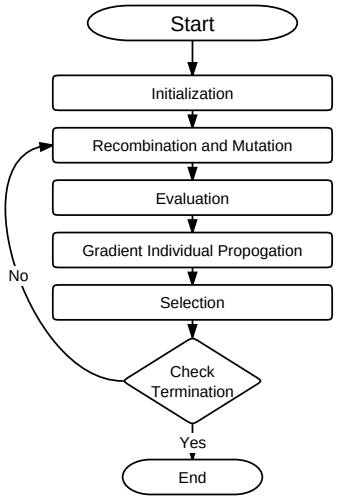


Figure 1: Flowchart for the hybrid evolution strategy algorithm

and the Hessian matrix is updated iteratively with information from new function and gradient evaluations. In fact, the original gradient individual/evolution strategy hybrid approaches of Tahk, et al [9, 10] used the quasi-Newton method to advance the gradient individual. In contrast, in this proposed approach, the Hessian matrix is completely recomputed at each generation from a local approximation to the objective function and should be more accurate than traditional quasi-Newton Hessian approximations.

3.Covariance Matrix Adaptation Evolution Strategy with Gradient Individual

The basic outline of the hybrid evolution strategy algorithm is illustrated in the flowchart in Figure 1. To study the efficacy of this version of the hybrid gradient individual / evolution strategy approach we have implemented the algorithm in the context of CMAES. While we have also done this in the context of the standard evolution strategy, we use CMAES here because it seems to have better global convergence properties. We want to see if the addition of the gradient individual to the offspring at each generation will negatively affect the convergence of the CMAES. In particular, we utilize the Matlab version of CMAES (version 2.54) provided publically by Hansen [6].

CMAES is an evolution strategy that adapts a full covariance matrix of a normal search distribution [5]. The strategy begins with an initial population of λ individuals $\mathbf{x}_{k=1:\lambda}^{(0)}$. After evaluating the objective function, the best μ individuals are selected as parents and their centroid is computed by using a weighted average: $\langle \mathbf{x} \rangle_W^{(0)} = \sum_{k=1}^{\mu} w_k \mathbf{x}_{k:\lambda}^{(0)}$, where the weights, w_i, are positive reals and sum to one. The notation $\mathbf{x}_{k:\lambda}$ is called selection notation and represents the point with the k^{th} lowest corresponding objective fitness value. While many weighting schemes have been proposed, here we use the super-linear weights: $w_i = \ln(\mu) - \ln(i), i = 1, \ldots, \mu$, wherein the individuals with lowest fitness values get the highest recombination weights.

After selection and recombination a new population is created by:

$$\mathbf{x}_{k=1:\lambda}^{(t+1)} = \langle \mathbf{x} \rangle_W^{(t)} + \sigma^{(t)} \mathbf{B}^{(t)} \mathbf{D}^{(t)} \mathbf{z}_{k=1:\lambda} \tag{8}$$

where $\mathbf{z}_k \sim N(0, \mathbf{I})$ are independent realizations of an n-dimensional standard normal distribution with mean zero and covariance matrix $\mathbf{I}$. The base points, $\mathbf{z}_k$, are rotated and scaled by the eigenvectors, $\mathbf{B}^{(t)}$, and the square root of the eigenvalues , $\mathbf{D}^{(t)}$, of the covariance matrix , $\mathbf{C}^{(t)}$. The covariance matrix, $\mathbf{C}^{(t)}$, and global step size, $\sigma^{(t)}$, are updated after each generation. This approach yields a strategy that is invariant to any linear transformation of the search space. Equations for initializing and updating the strategy parameters are given in [4].

Table 1: Pseudo-code of the hybrid algorithm.

Generate and evaluate initial population $\mathbf{x}_{k=1:\lambda}^{(0)}$
Append these points to the database
Choose best individual and set $x_g^{(0)}$
Set $t = 0$, $g_0 = 0_{n\times 1}, H_0 = C^{(0)} = I_{n\times n}$
repeat
  Compute nearest neighbor RBF and find g_t and H_t
  Update the gradient individual $x_g^(t+1) = x_g - (H_t)^{-1} g_t$
  Evaluate $x_g^{(t+1)}$ and append to the database if feasible
  Select the best μ individuals from the population and $x_g^{(t+1)}$
  Generate new population $\mathbf{x}_{k=1:\lambda}^{(t+1)}$ by recombination and mutation
  Evaluate new population and append to database.
  if $\min_i f(x_i) < f(x_g^{(t)})$ then
    swap individual with lowest function value and $x_g^{(t)}$
  end if
  $t = t + 1$
until Stopping criteria are satisfied

Pseudo-code for the CMAES-RBFGI algorithm is shown in Table 1. One additional feature that has not been mentioned is that a weighted norm is used to compute nearest neighbors for determining the support of the local radial basis function approximation. We use the current covariance matrix of the CMAES since it should reflect the shape of search distribution and the objective function surface. The eigenvector/eigenvalue decomposition of the current covariance matrix is $C = BD^2B^T$. The distance between a point $x \in \mathbb{R}^n$ and the current gradient individual x_g is measured as $\|(BD)^{-1}(x - x_g)\|_2$. (For instance, a unit ball in this norm will be elliptically shaped to fit in a long narrow valley in the search space.) The nearest neighbors in this norm should be ideal points for constructing an approximation to the Hessian matrix.

4. Hybrid Algorithm applied to test suite

A small suite of test problems has been selected to compare the performance of our hybrid CMAES gradient individual algorithm (CMAES-RBFGI) with ordinary CMAES. For comparison, we have also implemented the quasi-Newton Hessian-approximation gradient individual approach of Tahk, et al. [10] in the context of CMAES (their implementation was in the standard evolution strategy); we refer to this implementation as CMAES-QNGI.

In CMAES-QNGI, after recombination (finding the centroid of the selected parents), only the first half of the new population of individuals is generated by Eq.8. After evaluating the objective function for these individuals, the current gradient individual is swapped for an individual with lower objective function value, if one exists, to ensure that the gradient individual is the current best point. The remainder of the current population is formed by reflecting the first half of the population symmetrical through the gradient individual in $\mathbb{R}^n$. This symmetrically selected population reduces the order of the discretization errors in forming the first and second derivative approximations at the gradient individual.

A summary of the selected test functions is shown in Table 2. Functions f_1, the classical sphere function, and f_2 are simple unimodal functions. f_3 is a simple unimodal function with random noise. f_4 is a simple unimodal function but is not differentiable at the minimum (or at many other points). f_5 is the classical Rosenbrock function which has two minima and a long flat valley. f_6, f_7, and f_8 are all multimodal. The Rastrigin function, f_7 presents some difficulty for CMAES, while Ackley function has the feature that it is not differentiable at the global minimum. All of the functions have their global minimum value of zero at zero.

Table 2: Test Functions for Hybrid Optimization Strategy

Name	Definition	Search Domain
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-40, 60]^n$
Schwefel 1.2	$f_2(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-40, 60]^n$
Schwefel 1.2 with noise	$f_3(x) = f_2(x)(1 + 0.4z)$ where z is $N(0,1)^n$	$[-40, 60]^n$
Schwefel 1.5	$f_4(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-40, 60]^n$
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	$[-40, 60]^n$
Griewank	$f_6(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right)$	$[-600, 600]^n$
Rastrigin	$f_7(x) = 10n + \sum_{i=1}^{n} \sum_{i=1}^{n} \left(x_i^2 - 10 \cos(2\pi x_i) \right)$	$[-40, 60]^n$
Ackley	$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^n$

Figure 2 shows convergence graphs for each of the test functions for each of the three algorithms: CMAES, CMAES-RBFGI, CMAES-QNGI. For each function the dimension is set at $n = 10$. The population size is $\lambda = 30$ with $\mu = 15$ parents being selected at each generation. The initial global step size, σ is set to 30% of the total length of the search domain in each dimension. The initial population is sampled from a uniform distribution, and the same samples are used to initialize each of the three algorithms. For the CMAES-RBFGI algorithm, the local RBF approximation is constructed using the k nearest neighbors with $k = \lceil 1.5(n+1)(n+2)/2 \rceil = 99$ which is just 50% larger than the minimum number of points necessary to construct a quadratic polynomial interpolant in $\mathbb{R}^n$. The k nearest neighbors are selected from among the last $N = 2k = 198$ individuals that have been evaluated. For each algorithm, 30 trials are conducted for each test function. The algorithm is stopped when a minimum objective function value of 10^{-10} is reached or when the best objective function value does not change for the last $10 + 30n/\lambda$ generations or when the ratio of the range of the current function values to the maximum current function value is below TolFun$= 5 \times 10^{-10}$.

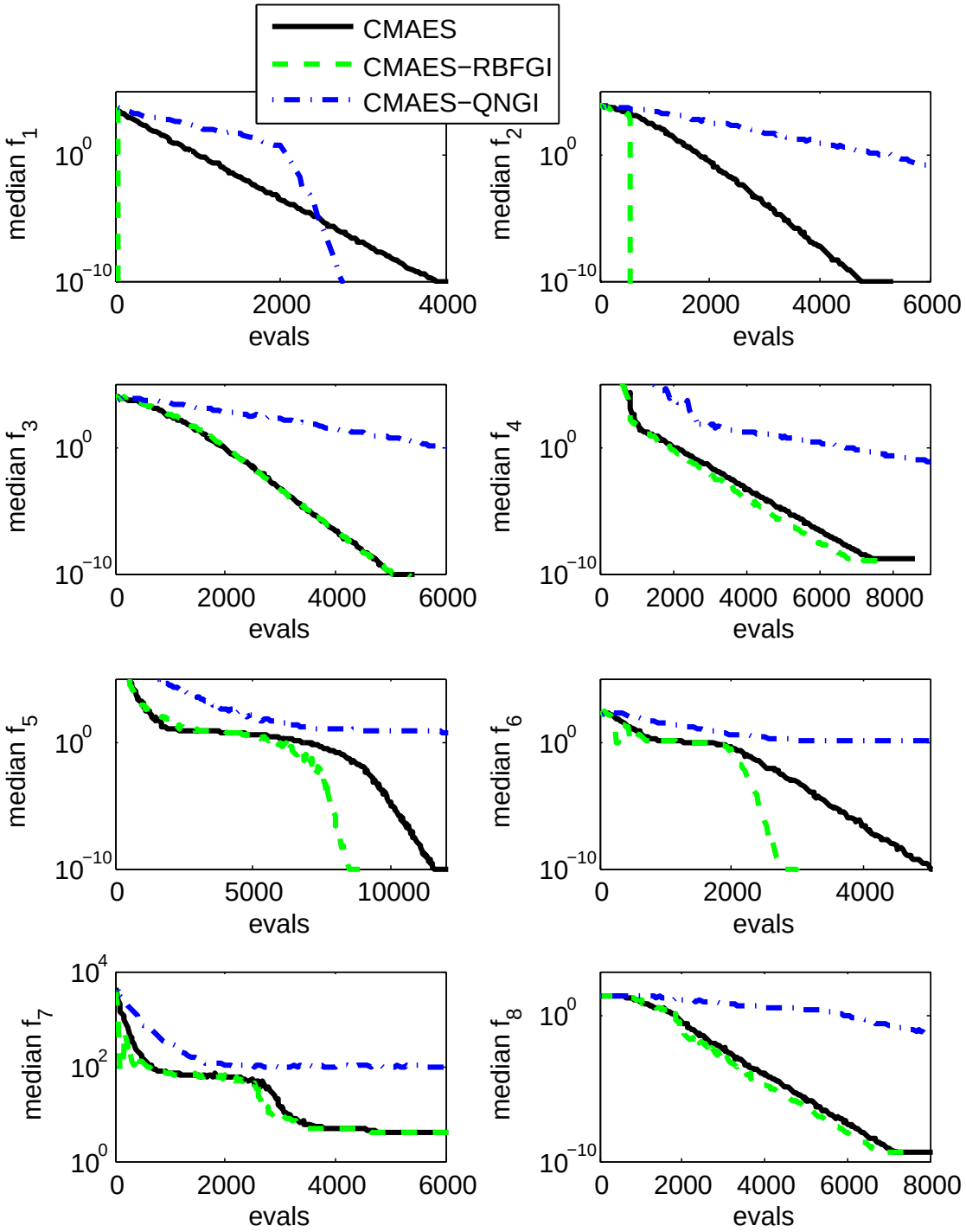


Figure 2: Convergence of the median objective value over 30 trials for 10 dimensional test functions.

The convergence graphs shown in Figure 2 illustrate several things. First, the approach of Tahk, et al. [9, 10] appears to interfere with the convergence of CMAES. This does not seem to be a case of simple error in implementing their strategy as we have been able to successfully implement and test it in the context of a standard evolution strategy. Rather the symmetrization of the population at each generation seems to interfere with the finely tuned covariance matrix adaptation and step size adaptation algorithm in this version of CMAES. The RBF gradient individual approach does not have this difficulty since it does not modify the existing population of the evolution strategy in any way, but simply appends an extra individual to the population. Secondly, for simple, smooth convex functions the CMAES-RBFGI

algorithm is greatly accelerated relative to CMAES as can be seen for functions f_1 : Sphere and f_2 : Schwefel 1.2. When there is strong noise in the objective function we would not expect a gradient-based search to perform well; as can be seen in with f_3, there is no acceleration with CMAES-RBFGI. Functions f_4 : Schwefel 1.5 (unimodal) and f_8 : Ackley (multimodal) are not differentiable at the global minimum (both have a sharp points). As can be seen in the figure, little acceleration is achieved by CMAES-RBFGI. (In a test not shown here, the Ackley function was squared to remove the singularity in the first derivative and the result appeared more like that of f_6.) For the Rosenbrock function, f_5 and the Griewank function, f_6 the convergence graphs show that the convergence of CMAES-RBFGI accelerates dramatically as the population approaches the minimizer. Finally, for the Rastrigin function, f_7 there is very little difference between CMAES and CMAES-RBFGI because both algorithms have difficulty locating the global minimum. Instead they converge to a variety of local minima.
As for reliability, the algorithms are nearly indistiguishable. For functions $f_1 - -f_4$ both CMAES and CMAES-RBFGI found the global optimum for all trials. For the Rosenbrock function, f_5 both algorithms located the global optimum in 28 of the 30 trials. For the Griewank function, f_6 CMAES found the global optimum 21 times while CMAES-RBF-GI found it 23 times. Neither algorithm ever found the global optimum for the Rastrigin function, f_7. For the Ackley function, f_8 CMAES only failed to find the global optimum once, while CMAES-RBFGI found it for all 30 trials. These results are promising as they show that the addition of the gradient individual does not impede the global search capability of the evolution strategy.

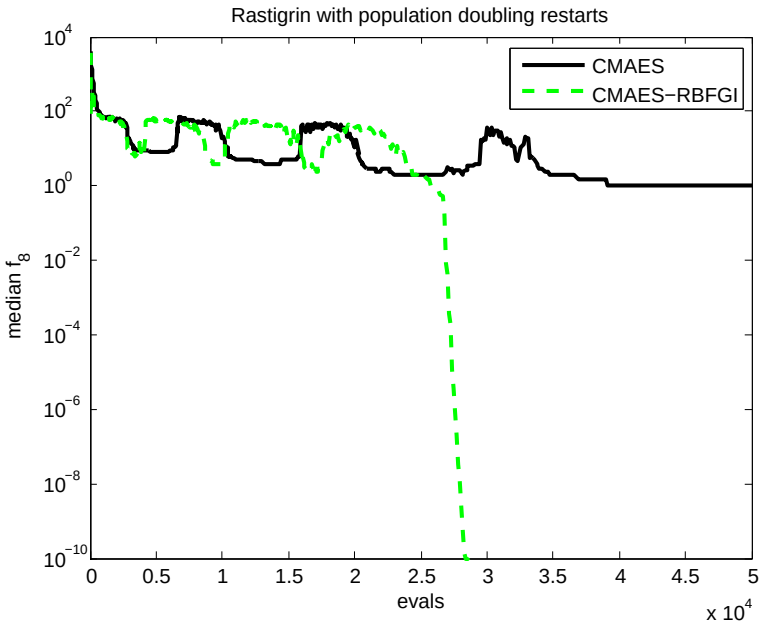


Figure 3: Convergence comparison for Rastrigin function with restarts

For the Rastrigin function we conduct a new experiment in which we run the evolution strategy using a population doubling restart strategy [1]. Each variation of CMAES is run until convergence as above with same $\lambda = 30$ as above and then the population size λ is doubled while $\mu = \lambda/2$ and the algorithm restarted. For CMAES-RBFGI the database of points is maintained between restarts. A total of 4 restarts are allowed to a maximum population size of 480. The maximum number of function evaluations remains capped at 50,000. The CMAES population size increasing strategy has been shown to be one of the most successful global optimization methods presently known for some benchmark problems [1]. The convergence graph for the median function value over 30 trials is shown in Figure 3. Most significantly, CMAES-RBFGI demonstrates far greater reliability. CMAES correctly locates, within the alloted 50,000 function evaluations, the global minimum at zero in 5 of the 30 trials, while CMAES-RBFGI finds the global minimum in 22 of the 30 trials.

5. Application to calibration of a watershed model.

To demonstrate the utility of the CMAES-RBFGI algorithm in a practical setting we used the algorithm to calibrate HYMOD, a five-parameter conceptual rainfall-runoff model (see Figure 5), introduced in [2].

In short, given time series of daily precipitation (P) and evapotranspiration (ET) data the objective is tune the parameters so that the least squares error between the model predicted stream flow time series and the observed stream flow time series is minimized. Such problems are usually characterized by multiple minima, sometimes unidentifiable parameters and even discontinuities in the objective function.

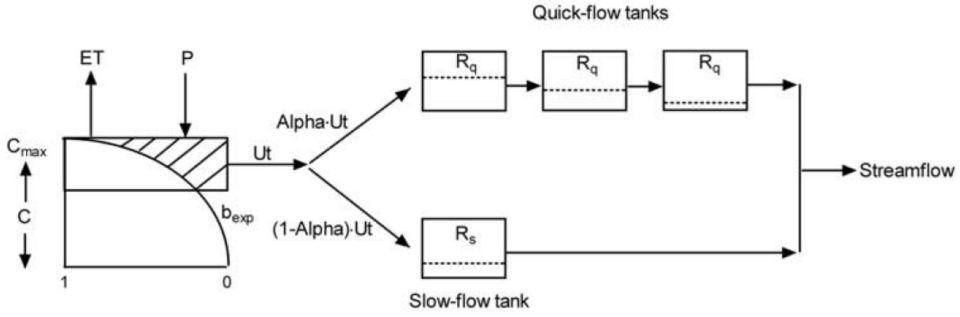


Figure 4: Schematic representation of the HYMOD model; from [12]

The HYMOD model is a simple rainfall excess model (details in [Moore 1985]), connected with two series of linear reservoirs: three identical quick and a single slow response reservoir. The five parameters to be calibrated for the model stream flow to match the observed stream flow data are: the maximum storage capacity of the catchement, $C_{\max}$; the degree of spatial variability of the soil moisture capacity, $b_{\exp}$; the factor distributing flow between the two series of reservoirs, Alpha; and the residence time of the linear quick and slow reservoirs, R_q and R_s, respectively.
Three years, October 1, 1948, to September 30, 1951, of daily hydrologic data from the Leaf River watershed were used for model calibration. This humid 1944 km^2 watershed is located north of Collins, Mississippi. The data, obtained from the National Weather Service Hydrology Labratory, consist of mean areal precipitation (mm/d), potential evapotranspiration (mm/d), and stream flow (m^3/s).

Table 3: Uncertainty ranges of HYMOD model parameters

	Minimum	Maximum	Unit
$C_{\max}$	1.000	500.000	mm
$b_{\exp}$	0.100	2.000	
Alpha	0.100	0.990	
R_s	0.000	0.300	day
R_q	0.000	0.990	day

The CMAES and CMAES-RBFGI algorithms are each applied to the optimization or calibration of the HYMOD model for 30 trials. The initial ranges of the parameter values are shown in Table 3. As for the test functions discussed above, the algorithms are initialized with the same uniform initial distributions. We set $\lambda = 10$ and $\mu = 5$. Each algorithm stops when TolFun$= 5e-4$, as described previously. For CMAES-RBFGI the number of nearest neighbors used for the local RBF is $k = \lceil 1.5(n+1)(n+2)/2 \rceil = 23$ chosen from the last $N = 2k = 46$ evaluated points.

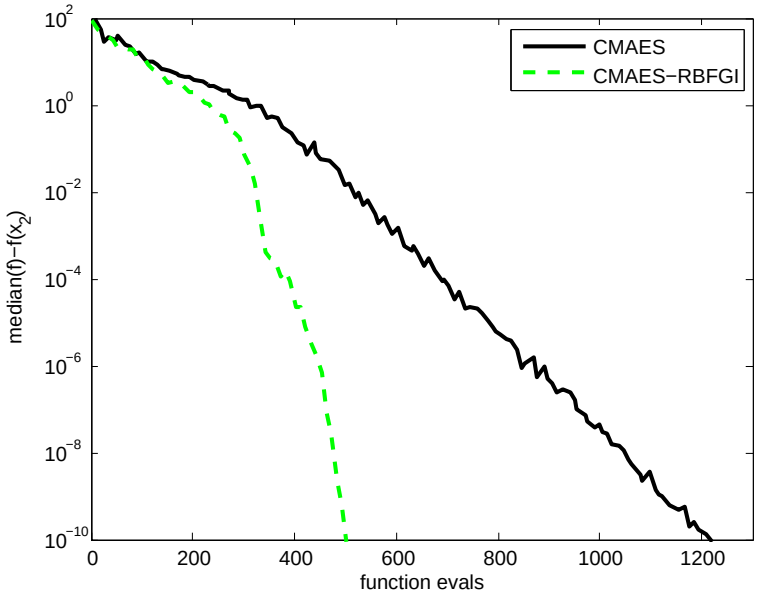


Figure 5: Convergence of CMAES and CMAES-RBFGI for the HYMOD model.

There are many local minima, but CMAES and CMAES-RBFGI nearly always converge to one of two minima $\mathbf{x}_1 = (157.0796, 0.5440, 0.2376, 0.2624, 0.8178)$ where $f_{\text{HYMOD}}(\mathbf{x}_1) = 128.5346$ or $\mathbf{x}_2 = (146.9868, 0.7165, 0.2416, 0.2619, 0.8313)$ where $f_{\text{HYMOD}}(\mathbf{x}_2) = 128.6374$. The global minimum appears to be at $\mathbf{x}_1$ but CMAES converges to $\mathbf{x_2}$ in 28 of 30 trials, while CMAES-RBFGI converges to the same minimum in 27 of 30 trials. Evidently, the basin of attraction for the global minimum, $\mathbf{x}_1$, is quite small as both algorithms have trouble finding it. The accelerated convergence of CMAES-RBF to the local minimum of the HYMOD model is demonstrated in Figure 5. For each trial the best function value is saved at each generation. The median function value over the thirty trials minus the value at the local minimum , $f(\mathbf{x_2})$, is plotted versus the number of function evaluations. As Fiure 5 demonstrates, the increase in convergence speed is quite dramatic: CMAES-RBFGI typically converges with fewer than half of the objective function evaluations. Though neither algorithm reliably locates the global minimum, both algorithms give good approximations to the global minimum that produce adequate approximations to the daily stream flows. To locate the global minimum reliably, a restart strategy could be used as with the Rastrigin function above. The RBFGI method would still accelerate the convergence significantly.

6. Conclusions

The gradient individual hybridization approach for evolution strategies has been shown to be effective for significantly accelerating the convergence of the covariance matrix adaptation evolution strategy. Likewise, it also works with the standard evolution strategy, though the results are not shown here. To develop a hybrid evolution strategy using local RBF approximation, as we have done here, requires very little modification of the actual evolution strategy. The evolving population itself is not modified, but the additional gradient individual is added at each generation. In the gradient individual approach of Tahk, et al the population is chosen symmetrically at each generation and, as seen here, this can interfere with the convergence of CMAES. Another advantage to this approach is that no minimum population size is required. In [9, 10] the population size must be at least twice the dimension of the search space to estimate the gradient vector. The downside of RBFGI approach is that it is expensive to form the coefficient matrix in Eq. 4. Moreover, the size of that system scales as $O(n^2)$, so its solution by a direct method requires $O(n^6)$ operations per generation. Thus, there is a trade-off between the computational complexity of the RBFGI method and the gain due to fewer function evaluations. For expensive objective functions the cost of adding a gradient individual propagated by local radial basis function approximation is expected to be incidental and the increase in speed can be enormous. As a result, the RBFGI approach should be incorporated into evolution strategies for expensive functions as it can sometimes greatly increase converge speed and reliability with little downside.

Acknowledgements

The first author gratefully acknowledges the support of ARO Grant W911NF-06-0306.

References

[**1**] A. Auger and N. Hansen, A restart CMA evolution strategy with increasing population size, *Proc. IEEE Congr. Evol. Comput.,* Edinburgh, U.K., vol. 1, 1777-1784, 2005.

[**2**] D.P. Boyle, Multicriteria calibration of hydrological models, Ph.D. disserations, Univ. Ariz., Tuscon, 200.

[**3**] B. Fornberg, N. Flyer, and J.M. Russell, Comparisons between pseudospectral and radial basis function derivative approximations, *IMA Journal of Numerical Analysis*, 30 (1), 149-172, 2009.

[**4**] N. Hansen and S. Kern, Evaluating the CMA strategy on multimodal test functions, *Lecture Notes in Computer Science, X*, Yao, et al Eds., Springer-Verlag, vol. 3242, Parallel Problem Solving from Nature – PPSN VIII, 282-291, 2004.

[**5**] N. Hansen and A. Ostermeier, Completely arbitrary normal mutation distributions in evolution strategies, *Proc. IEEE Congr. Evol. Comput.,* Nagoya, Japan, 312-317, 1996.

[**6**] N. Hansen, Source Code for CMA-ES, `http://www.lri.fr/~hansen/cmaes_inmatlab.html`.

[**7**] M.J.D. Powell, The theory of radial basis function approximation in 1990, *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*, W. Light Ed., London, U.K., Oxford Univ. Press, 105-210, 1992.

[**8**] R.G. Regis and C.A. Shoemaker, Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions, *IEEE Trans. Evol. Comput.,* 8 (5), 490-505, 2004.

[**9**] M.-J. Tahk, H.W. Woo, and M.S. Park, A hybrid optimization method of evolutionary and gradient search, *Engineering Optimization*, 39 (1), 87-104, 2007.

[**10**] M.-J. Tahk, M.S. Park, H.W. Woo, H.J. Kim, Hessian approximation algorithms for hybrid optimization methods, *Engineering Optimization*, 41 (7), 609-633, 2009.

[**11**] H.W. Woo, H.H. Kwon, and M.-J. Tahk, A hybrid method of evolutionary algorithms and gradient search, *Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, S.C. Mukhopadhyay and G. Sen Gupta Eds., Palmerston North, New Zealand, 2004 (ISBN 0-476-00994-4).

[**12**] J.A. Vrugt, H.V. Gupta, W. Bouten, and S. Sorooshian, A shuffled complex evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, *Water Resources Research*, 39 (8), doi:10.1029/2002WR001642, 2003.

Appendix B

Hybrid Optimization using an Evolutionary Strategy and Surrogate Assisted Local Search

Jeff S. Baggett and Brian E. Skahill

submitted to Engineering Opimization

October 2010.

Engineering Optimization
Vol. 00, No. 00, Month 200x, 1–18

RESEARCH ARTICLE

Hybrid Optimization using an Evolutionary Strategy and Surrogate Assisted Local Search

Jeff S. Baggett[a*] and Brian E. Skahill[b]

[a] *University of Wisconsin - La Crosse, La Crosse, WI, USA*; [b] *US Army Engineer Research and Development Center, Vicksburg, MS, USA*

(*September, 2010*)

We present a new derivative-free hybrid algorithm for global optimization of expensive black box functions. The algorithm uses an evolution strategy for global search. Convergence toward local minima is accelerated by including a local search individual in each generation. The local search individual is computed by extracting derivative information from a radial basis function approximation to the objective function interpolated from previously evaluated points in the evolutionary strategy. This hybrid approach does not require artificial or user-defined switching between global and local search. Numerical results are presented on mathematical test problems from the optimization literature and for a small dimensional conceptual watershed model from hydrology.

Keywords: Hybrid algorithms; Evolution strategies; Quasi-Newton method

1. Introduction

Evolution strategies are known to be reliable but expensive for approximating global optima particularly for multi-modal fitness functions. A large population size is required for the strategy to explore the real parameter space, but this slows local convergence. In contrast, classical gradient-based algorithms are exploitative in nature and converge quickly to local minima, but they are not good at finding the global minimum.

Many algorithms attempt to accelerate the convergence of the evolutionary strategy, or other population-based search method, by either switching to a local, usually gradient-based, search at some user-defined threshold or by applying some local search operator at every generation so that the global and local searches are interwoven. Examples of the former strategy include switching from particle swarm optimization to sequential quadratic programming at a user defined threshold (Min *et al.* 2007) and similarly switching from a genetic algorithm to a Levenberg-Marquardt algorithm (Peters *et al.* 2010). Examples

*Corresponding author. Email baggett.jeff@uwlax.edu

ISSN: 0305-215X print/ISSN 1029-0273 online

of interwoven strategies include incorporation of an extra individual at each generation of an evolution strategy calculated from an approximated Newton step (Woo *et al.* 2004, Tahk *et al.* 2007, 2009), the use of a discrete gradient operator to improve the best individual in each generation of an evolution strategy (Abbas *et al.* 2003), and a hybrid genetic algorithm that uses a quasi-Newton step to attempt to improve the fitness of every individual at each iteration (Renders and Flasse 1996).

Another approach taken to lessen the number of expensive objective function evaluations in evolutionary strategies and other population-based algorithms is to use function approximation models as surrogates for the objective function. Typically, estimated function values are used to screen offspring and the more expensive objective function is evaluated only at the most promising offspring (Regis and Shoemaker 2004, Kern *et al.* 2004). In this paper we propose to use similar function approximation models to approximate derivative information for a local search that will be interwoven with the global evolutionary strategy. In fact, the same function approximation model can be used to approximate derivatives for the local search and as a surrogate to screen objective function values for the global search, though we do not report on that here.

We follow essentially the same framework as the hybrid evolutionary strategy first outlined in (Woo *et al.* 2004) and subsequently improved in (Tahk *et al.* 2007, 2009). In that algorithm, a standard evolutionary strategy is used to advance the population at each generation. Additionally, an individual called the gradient individual is propogated alongside the evolving population. The gradient individual is calculated by making a Newton update from the gradient individual of the previous generation or from the fittest point of the current generation. The gradient is estimated from a least squares finite difference approximation and the Hessian is iteratively approximated by one-dimensional finite difference updates using the previously evaluated points in the population. The gradient individual hybrid approach has been shown to improve convergence of the standard evolutionary strategy on a small number of mathematical test problems. One drawback to the algorithm is that the individuals in the evolving population are selected symmetrically about the estimated minimum of the current generation: $x_{\min} \pm \Delta x$. Selecting the population symmetrically increases the accuracy of the finite difference discretizations of the derivatives, however the symmetric population seems to interfere (Baggett and Skahill 2010) with the convergence of the covariance matrix adaptation variation of the evolutionary strategy (CMAES due to Hansen and Ostermeier 1996). While we have not determined exactly how the symmetrized population interferes with adaptation in CMAES, we think it is likely related to the covariance matrix updating and/or the step length updating within that algorithm. Another drawback to the symmetrized finite difference hybrid algorithm is that the minimum population size for the evolutionary strategy is $2n$, where n is the dimension of the search space.

Our proposed algorithm, however, differs from the gradient individual hybrid evolution strategy (Woo *et al.* 2004, Tahk *et al.* 2007, 2009) in two important respects: first, instead of using finite differences to approximate local derivative information we fit a local function approximation model to previously evaluated points and differentiate the model analytically, and second, the evolution strategy is the covariance matrix adaptation evolution strategy (CMAES) (Hansen and Ostermeier 1996). The use of the surrogate model to approximate the derivatives allows the estimation of derivatives without making any modifications to the basic structure of the evolutionary strategy. In fact, this approach allows for hybridization of virtually any population-based search algorithm. In this article we will focus on hybridization of CMAES which is known to be particularly effective at approximating global optima. CMAES is especially effective at locating global optima

when used in conjunction with a population doubling strategy as described in Auger and Hansen (2005). More recently, a two population restart strategy in which one population grows exponentially larger while the other is maintained at a small size has shown promise (Hansen 2009), but is not explored here.

The derivative estimation method will be explained first and it will be shown how the approximated derivatives are used to perform local search by calculating a "local search individual" in each generation. Next the hybrid algorithm will be summarized. We have implemented the hybrid algorithm in CMAES, and it will be shown how the new hybrid algorithm performs on a small suite of test functions in 10 and 30 dimensions. Finally, we briefly demonstrate the calibration of a conceptual watershed model using the hybrid algorithm.

2. Local Search using Radial Basis Functions

The foundation of this optimization algorithm is an evolution strategy in which new offspring are produced at each generation by recombination and mutation (see Figure 2). The objective function is evaluated at each of these offspring and the fittest offspring are selected as parents for the next generation. In the hybrid approach, an additional individual, called the local search individual, is propagated by a different mechanism each generation. Tahk *et al.* (2007) refer to this additional individual as the gradient individual, but we call it the local search individual since in practice it could be the result of *any* local search that is a function of the previously evaluated points. The current local search individual, x^t_{ls}, is either the fittest offspring of the current generation (individual with lowest function value) or the local search individual from the previous generation. From information gathered during the evolution of the population the first and second derivatives of the objective function are estimated at x^t_{ls} and used to perform an update of the local search individual which is hopefully moved closer to a stationary point.

As an evolution strategy proceeds, it typically does not use the previously evaluated points beyond the current generation; however, in our hybrid strategy we store the last N points and their evaluated functions values in a database. In practice, if the objective function is very expensive to evaluate, we might use *all* of the previously evaluated points. To update the local search individual we use a k-nearest neighbor local function approximation of the objective function using the k nearest neighbors (Euclidean distance) of x^t_{ls} in the database to construct a cubic radial basis function (RBF) approximation:

$$s(x) = \sum_{i=1}^{k} w_i \phi\left(\|x - x_i\|_2\right) + p(x), \quad x \in \mathbb{R}^n \tag{1}$$

where $x_i, i = 1, 2, \ldots, k$ are the k nearest neighbors of x^t_{ls} in the n-dimensional search space, p is in Π^n_2 (the linear space of polynomials in n variables of degree less than or equal to 2), and $\phi(r) = r^3$. Cubic radial basis functions were selected not only for their simplicity and differentiability, but also because they have been used successfully as surrogate models for pre-evaluating function values to lessen the number of function evaluations required by an evolution strategy (Regis and Shoemaker 2004).

Define the matrix $\Phi \in \mathbb{R}^{k\times k}$ by

$$(\Phi)_{ij} := \phi\left(\|x_i - x_j\|_2\right), \quad i, j = 1, \ldots k. \tag{2}$$

Let $\hat{n} = (n+1)(n+2)/2$ be the dimension of Π_2^n, let $p_1, \ldots, p_{\hat{n}}$ be a basis of this linear space, and define the matrix $P \in \mathbb{R}^{k \times \hat{n}}$ as follows:

$$P_{ij} := p_j(x_i), \quad i = 1, \ldots, k; j = 1, \ldots, \hat{n}. \tag{3}$$

In this model, the RBF that interpolates the points $(x_1, f(x_1)), \ldots (x_k, f(x_k))$ is obtained by solving the system

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} w \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0_{\hat{n}} \end{pmatrix} \tag{4}$$

where $F = (f(x_1), \ldots, f(x_k))^T, w = (w_1, \ldots, w_k) \in \mathbb{R}^k$ and $c = (c_1, \ldots, c_{\hat{n}})^T \in \mathbb{R}^{\hat{n}}$. Powell (1992) gives sufficient and necessary conditions for the system above to be uniquely solvable, but in practice the real issue can be that the coefficient matrix above becomes ill-conditioned. However, we have found that simply rescaling and shifting the points $x_1, \ldots, x_k$ so that they all lie in $[-1, 1]^n$ is usually sufficient to address this issue.

Once the RBF, $s(x)$, has been determined by Eq.(1), then $s(x)$ is differentiated analytically to determine approximations to the gradient and Hessian of the objective function, $f(x)$. For the gradient vector, g, evaluated at the local search individual, x_{ls}^t:

$$g_i = \left(\nabla f(x_{\text{ls}}^t)\right)_i = \frac{\partial}{\partial x_i} f(x_{\text{ls}}^t) \approx \left(\nabla s(x_{\text{ls}}^t)\right)_i = \frac{\partial}{\partial x_i} s(x_{\text{ls}}^t), \quad i = 1, \ldots, n \tag{5}$$

For the Hessian matrix, H, evaluated at the gradient individual, x_{ls}^t:

$$H_{ij} = \left(H(x_{\text{ls}}^t)_{ij}\right) = \frac{\partial^2}{\partial x_i \partial x_j} f(x_{\text{ls}}^t) \approx \frac{\partial^2}{\partial x_i \partial x_j} s(x_{\text{ls}}^t), \quad i, j = 1, \ldots, n \tag{6}$$

Similar techniques for derivative approximation are routinely used in the numerical solution of partial differential equations using so-called meshless methods. Moreover, such approximations can be spectrally accurate (faster than polynomial in the grid size) depending on the selection of interpolation points (Fornberg *et al.* 2009).

Once the offspring and their function values from the current generation have been appended to the database, we construct the RBF as above and determine the gradient and Hessian approximated at the current local search individual x_{ls}^t. Finally, a new local search individual is found by the standard update:

$$x_{\text{ls}}^{t+1} = x_{\text{ls}}^t - H^{-1} g. \tag{7}$$

The new local search individual is then added to the current generation of offspring for possible selection. When the population, and hence the points in the database, are sufficiently close to a minimum then the gradient and Hessian can be accurate and can yield fast local convergence to the minimum. However, when the evolving population is far from a local minimum, then the gradient and Hessian tend to be inaccurate and the update does not lead to a minimum. In such cases, x_{ls}^{t+1}, is typically not selected as an offspring by the evolutionary strategy and thus does no harm to convergence of the gloal search.

This method of gradient-based search is similar to a quasi-Newton method, but in quasi-Newton methods the first derivatives of the function are usually available (or ap-

proximated by centered finite differences), and the Hessian matrix is updated iteratively with information from new function and gradient evaluations. In fact, the original evolution strategy hybrid approaches of Tahk *et al.* (2007, 2009) used the quasi-Newton method to advance their gradient individual. In contrast, in this proposed approach, the Hessian matrix is completely recomputed at each generation from a local approximation to the objective function. In the next subsection we give a synthetic numerical comparison to demonstrate that the numerical accuracy of our method and that of Tahk *et al.* (2009) are comparable. The main difference in the methods is that, due to the use of the function approximation model for derivative calculation, our algorithm does not require symmetrizing the population and thus can be used to estimate derivatives and add local search to any population-based search algorithm.

The local search technique proposed here could easily be modified so that the local search individual is not simply the approximate Newton point. For instance, because we have the local function approximation available, the local search individual could be the result of a trust region search of the radial basis function along the Newton update direction similar to the approach described by Wild *et al.* (2008). In fact, any kind of local search algorithm that utilized primarily the previously collected points could be used to find the local search individual.

2.1. *A numerical comparison*

To demonstrate the capability of the radial basis function approach to approximating derivatives the 10-dimensional generalized Rosenbrock function (see Table 2) is used as a test-case. To facilitate a direct comparision with the Hessian approximation approach described by Tahk *et al.* (2009) the following synthetic numerical experiment was conducted.

The global minimizer of the Rosenbrock function is the point $\mathbf{1}$ consisting of all ones. A synthetic sequence of points was selected that converges toward the global minimizer at a constant rate:

$$x^{(i)} = \mathbf{1} + 0.01\frac{100-i}{100}(1,2,3,4,5,6,7,8,9,10), \qquad i = 1,\ldots,100 \tag{8}$$

At each iteration a population, from a normal distribution as in an evolution strategy, of $\lambda = 80$ points is generated around $x^{(i)}$ in two steps. In the first step, $\mu = 40$ points are generated by:

$$x = x^{(i)} + 0.01z, \tag{9}$$

where z is selected from the 10-dimensional standard normal distribution with mean 0 and variance 1. In the second step, 40 additional points are generated by reflecting the first 40 points through the center point $x^{(i)}$ - this center point will serve as the linearization point for each iteration and the method proposed by Tahk *et al.* (2009) utilizes the reflected points to produce higher order approximations to the gradient and Hessian. The Rosenbrock function is evaluated at the 81 points and the points and function values are stored in a database for use by our radial basis function approach.

We do not present the details of the approach of Tahk *et al.* (2009) here, but at each iteration the gradient is approximated by a least squares fit of the centered finite differences of the symmetric pairs through the linearization point. Moreover, the inverse Hessian is sequentially updated with the centered difference second derivative information

from each symmetric pair of points. Utilizing these approximation techniques, at each iteration we obtain an approximation to the gradient and inverse Hessian: g_{Tahk} and H^{-1}_{Tahk}, respectively.

At each iteration two different approximations to the stationary point are computed. The first is the from the approximation due to Tahk *et al.* (2009):

$$x^{(i)}_{\text{Tahk}} = x^{(i)} + h_{\text{Tahk}} = x^{(i)} - H^{-1}_{\text{Tahk}} g_{\text{Tahk}}. \tag{10}$$

The second approximation uses the nearest $k = 132$ points from among the last $N = 264$ points stored in the database to compute the radial basis function approximations to the gradient and Hessian approximations described in equations (5) and (6); we label these g_{rbf} and H_{rbf}, respectively.

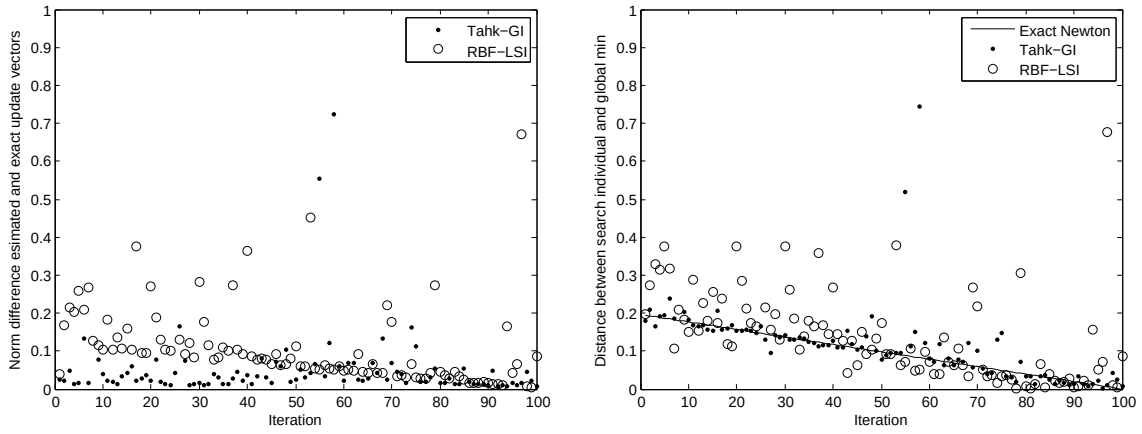
$$x^{(i)}_{\text{rbf}} = x^{(i)} + h_{\text{rbf}} = x^{(i)} - H^{-1}_{\text{rbf}} g_{\text{rbf}}. \tag{11}$$

We compare the update vectors h_{Tahk} and h_{rbf} to the analytically computed update vector in Figure 1(a). For the first 50 iterations the h_{Tahk} is a better approximation to the analytically computed update vector than computed by radial basis functions, but for iterations 51–100 the approximations are comparable. Even though h_{rbf} is not always the most accurate approximation to the update vector, it turns out that it does provide a good search direction. Figure 1(b) shows the distance from each of the approximated stationary points, $x^{(i)}_{\text{Tahk}}$ and $x^{(i)}_{\text{rbf}}$, as well as the analytically computed Newton point, to the global minimum of the 10-dimensional Rosenbrock function. It can be seen that the radial basis function approximation local search individuals and the Tahk *et al.* (2009) gradient individuals give comparable approximations to the global minimizer in spite of the fact that h_{rbf} is sometimes a less accurate approximation to the true update vector (from the analytic derivatives) than that of h_{Tahk}. In both plots, it can be seen that the radial basis function local search individiual is occasionally quite a bad approximation to the global minimum. This appears to happen because the problem of radial basis function interpolation and derivative interpolation is ill-conditioned and very sensitive to the choice of interpolation points. In most generations our simple choice of nearest neigbor points is adequate, but not always. This will be the subject of future work.

The radial basis function approach to computing the local search individual is the more expensive algorithm, but can be adapted to be used with any population based algorithm as it requires no modification of the underlying sampling algorithm. This makes the radial basis function approach suitable to be used with the covariance matrix adaptation evolution strategy as will be described in the next section.

3. Covariance Matrix Adaptation Evolution Strategy with Local Search Individual

The basic outline of the hybrid evolution strategy algorithm is illustrated by the flowchart in Figure 2. To study the efficacy of this version of our hybridized approach we have implemented the algorithm in the context of CMAES. While we have also done this in the context of the standard evolution strategy, we use CMAES here because it seems to have better global convergence properties (Hansen and Kern 2004, Hansen 2009). In particular, we utilize the Matlab version of CMAES (version 2.54) provided publically



(a) (b)

Figure 1. Comparison of rbf local search and Tahk local search for 10-dimensional Rosenbrock function. Figure (a) compares the update vectors for the two methods; Figure (b) shows that the distance from the approximated stationary point to the global minimizer is comparable for the two methods.

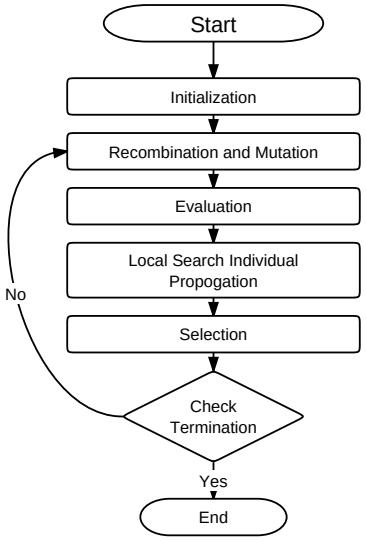


Figure 2. Flowchart for the hybrid evolution strategy algorithm

by Hansen (2004).

CMAES is an evolution strategy that adapts a full covariance matrix of a normal search distribution (Hansen and Ostermeier 1996). The strategy begins with an initial population of λ individuals $\mathbf{x}_{k=1:\lambda}^{(0)}$. After evaluating the objective function, the best μ individuals are selected as parents and their centroid is computed by using a weighted average: $\langle \mathbf{x} \rangle_W^{(0)} = \sum_{k=1}^{\mu} w_k \mathbf{x}_{k:\lambda}^{(0)}$, where the weights, w_i, are positive reals and sum to one. The notation $\mathbf{x}_{k:\lambda}$ is called selection notation and represents the point with the k^{th} lowest corresponding objective fitness value. While many weighting schemes have been proposed, here we use the super-linear weights: $w_i = \ln(\mu) - \ln(i), i = 1, \ldots, \mu$, wherein the individuals with lowest fitness values get the highest recombination weights.

After selection and recombination a new population is created by:

$$\mathbf{x}_{k=1:\lambda}^{(t+1)} = \langle \mathbf{x} \rangle_W^{(t)} + \sigma^{(t)} \mathbf{B}^{(t)} \mathbf{D}^{(t)} \mathbf{z}_{k=1:\lambda} \tag{12}$$

where $\mathbf{z}_k \sim N(0, \mathbf{I})$ are independent realizations of an n-dimensional standard normal distribution with mean zero and covariance matrix $\mathbf{I}$. The base points, $\mathbf{z}_k$, are rotated and scaled by the eigenvectors, $\mathbf{B}^{(t)}$, and the square root of the eigenvalues ,$\mathbf{D}^{(t)}$, of the covariance matrix ,$\mathbf{C}^{(t)}$. The covariance matrix, $\mathbf{C}^{(t)}$, and global step size, $\sigma^{(t)}$, are updated after each generation. This approach yields a strategy that is invariant to any linear transformation of the search space. Equations for initializing and updating the strategy parameters are given in (Hansen and Kern 2004). For complete details on the CMAES algorithm the tutorial (Hansen 2010) is a definitive source.

Table 1. Pseudo-code of the hybrid algorithm.

```
Generate and evaluate initial population x_{k=1:λ}^{(0)}
Append these points to the database
Choose best individual and set x_ls^{(0)}
Set t = 0, g_0 = 0_{n×1}, H_0 = C^{(0)} = I_{n×n}
repeat
   Compute nearest neighbor RBF and find g_t and H_t
   Update the local search individual x_ls^{(t+1)} = x_ls − (H_t)^{-1} g_t
   Evaluate x_ls^{(t+1)} and append to the database if feasible
   Select the best μ individuals from the population and x_ls^{(t+1)}
   Generate new population x_{k=1:λ}^{(t+1)} by recombination and mutation
   Evaluate new population and append to database.
   if min_i f(x_i) < f(x_ls^{(t)}) then
      swap individual with lowest function value and x_ls^{(t)}
   end if
   t = t + 1
until Stopping criteria are satisfied
```

Pseudo-code for the CMAES-RBFLSI algorithm is shown in Table 1. One additional feature that has not been mentioned is that a weighted norm is used to compute nearest neighbors for determining the support of the local radial basis function approximation. We use the current covariance matrix of the CMAES since it should reflect the shape of the search distribution and the objective function surface. The eigenvector/eigenvalue decomposition of the current covariance matrix is $C = BD^2B^T$. The distance between a point $x \in \mathbb{R}^n$ and the current gradient individual x_{ls} is measured as $\|(BD)^{-1}(x - x_{\text{ls}})\|_2$. (For instance, a unit ball in this norm will be elliptically shaped to fit in a long narrow valley in the search space.) The nearest neighbors in this norm should be ideal points for constructing an approximation to the Hessian matrix.

4. Hybrid Algorithm applied to mathematical test suite

A small suite of test problems has been selected to compare the performance of our hybrid CMAES local search individual algorithm (CMAES-RBFLSI) with ordinary CMAES.

Our aim is to establish that the hybridized approach does not change the global convergence properties of CMAES while accelerating the local convergence rate. For comparison, we have also implemented the quasi-Newton Hessian-approximation gradient individual approach of Tahk *et al.* (2009) in the context of CMAES (their implementation was in the standard evolution strategy); we refer to this implementation as CMAES-QNGI.

In CMAES-QNGI, after recombination (finding the centroid of the selected parents), only the first half of the new population of individuals is generated by equation (12). After evaluating the objective function for these individuals, the current gradient individual is swapped for an individual with lower objective function value, if one exists, to ensure that the gradient individual is the current best point. The remainder of the current population is formed by reflecting the first half of the population symmetrically through the gradient individual in $\mathbb{R}^n$. This symmetrically selected population reduces the order of the discretization errors in forming the first and second derivative approximations at the gradient individual.

A summary of the selected test functions is shown in Table 2. Function f_1 is the quadratic Schwefel 1.2 function and is a test case for CMAES because its elliptical contours test the ability of the algorithm to adapt the shape of the search distribution. f_2 is the cone function selected for its lack of differentiability at the minimum. f_3 is the classical generalized Rosenbrock function which has two minima for $n \geq 4$ and the long narrow valley which slows convergence for many algorithms. f_4 is the Schwefel 1.5 function which, while unimodal, is not differentiable if any $x_i = 0, i = 1, \ldots, n$. The Griewank function f_5 is multimodal, but is not particularly challenging for CMAES and is smooth. The Rastrigin function, f_6, is multimodal and smooth and is a difficult problem for CMAES (Auger and Hansen 2005). Finally, the Ackley function, f_7, is multimodal and smooth except at the minimum and is somewhat challenging in larger dimensions for CMAES.

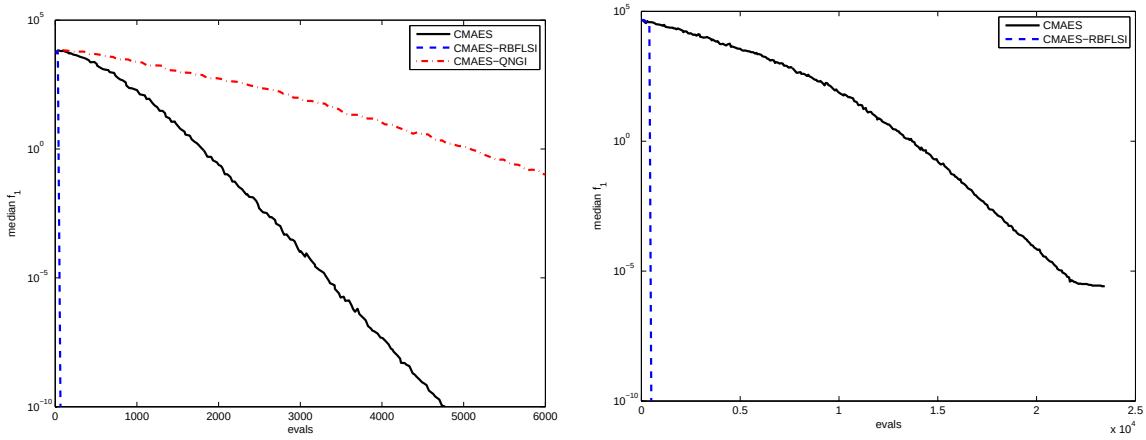
Table 2. Test Functions for Hybrid Optimization Strategy

Name	Definition	Search Domain
Schwefel 1.2	$f_1(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-40, 60]^n$
Cone	$f_2(x) = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$	$[-40, 60]^n$
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	$[-40, 60]^n$
Schwefel 1.5	$f_4(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-40, 60]^n$
Griewank	$f_5(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right)$	$[-600, 600]^n$
Rastrigin	$f_6(x) = 10n + \sum_{i=1}^{n} \sum_{i=1}^{n} \left(x_i^2 - 10 \cos(2\pi x_i) \right)$	$[-40, 60]^n$
Ackley	$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^n$

We examine only the convergence graphs to compare the algorithms. The CMAES-RBFLSI algorithm is expensive since it constructs the local radial basis function approximation at each generation based on $O(n^2)$ points using a naive linear solver that requires $O(n^6)$ operations. The solution time of (4) can be improved by using a null space method or iterative methods, but the presumption here is that the objective function evaluations are very expensive. Thus local search algorithms, even expensive ones, can and should be used to improve the efficiency of the overall search.

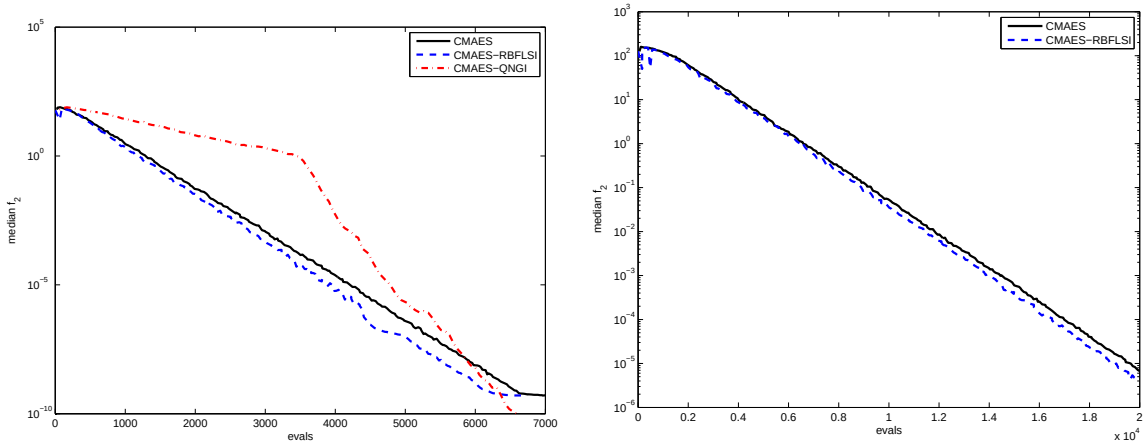
Figures 3-9 shows median convergence graphs based on a set of 30 trials for each of the test functions for each of the three algorithms: CMAES, CMAES-RBFGI, CMAES-QNGI. In each figure the results for dimension $n = 10$ are shown in plot (a), and the results for dimension $n = 30$ are shown in plot (b). We do not show the results of CMAES-QNGI at $n = 30$ since that algorithm does not perform well as will be discussed further below. For $n = 10$, the population size $\lambda = 30$ with $\mu = 15$ parents being selected at each generation, while with $n = 30$ we use $\lambda = 80$ and $\mu = 40$.. The initial global step size, σ is set to 30% of the total length of the search domain in each dimension. The initial population is sampled from a uniform distribution, and the same samples are used to initialize each of the three algorithms. For the CMAES-RBFLSI algorithm, the local RBF approximation is constructed using the k nearest neighbors with $k = (n + 1)(n + 2)$ which is twice the minimum number of points necessary to construct a quadratic polynomial interpolant in $\mathbb{R}^n$. The k nearest neighbors are selected from among the last $N = 2k$ individuals that have been evaluated. The algorithm is stopped when a minimum objective function value of 10^{-10} is reached or when the best objective function value does not change for the last $10 + 30n/\lambda$ generations or when the ratio of the range of the current function values to the maximum current function value is below TolFun$= 5 \times 10^{-10}$. The maximum number of function evaluations was set to be $n \times 10^4$.

Several points can be made upon inspection of the convergence graphs in Figures 3-9. First, the approach of Tahk *et al.* (2009) appears to interfere with the convergence of



(a) (b)

Figure 3. Convergence graphs for, f_1, the Schwefel 1.2 quadratic, unimodal function in 10 (a) and 30 (b) dimensions. Convergence is nearly instantaneous for CMAES-RBFLSI once enough points are obtained for an interpolant.

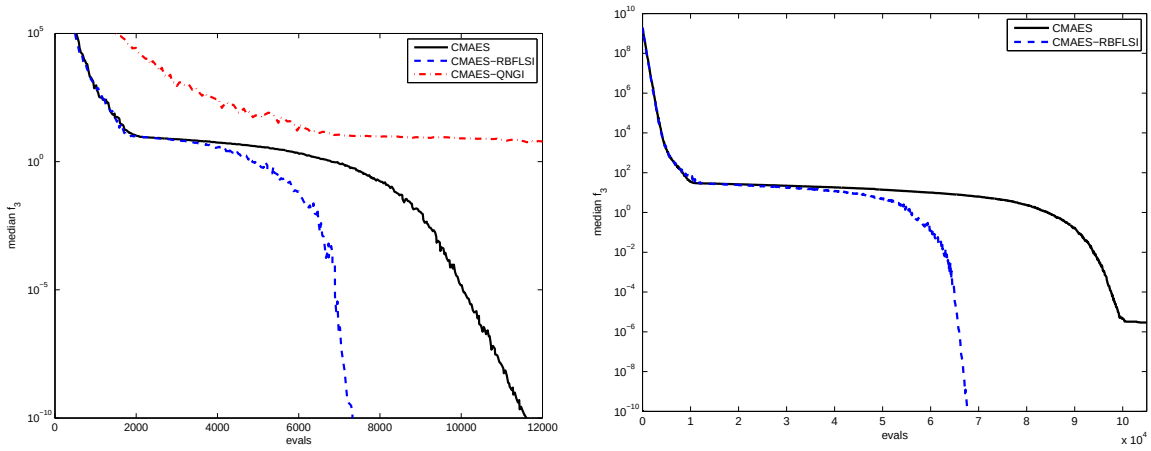


(a) (b)

Figure 4. Convergence graphs for, f_2, the cone function in 10 (a) and 30 (b) dimensions. The cone function is unimodal, but not differntiable at the minimum. Speedup is neglible.

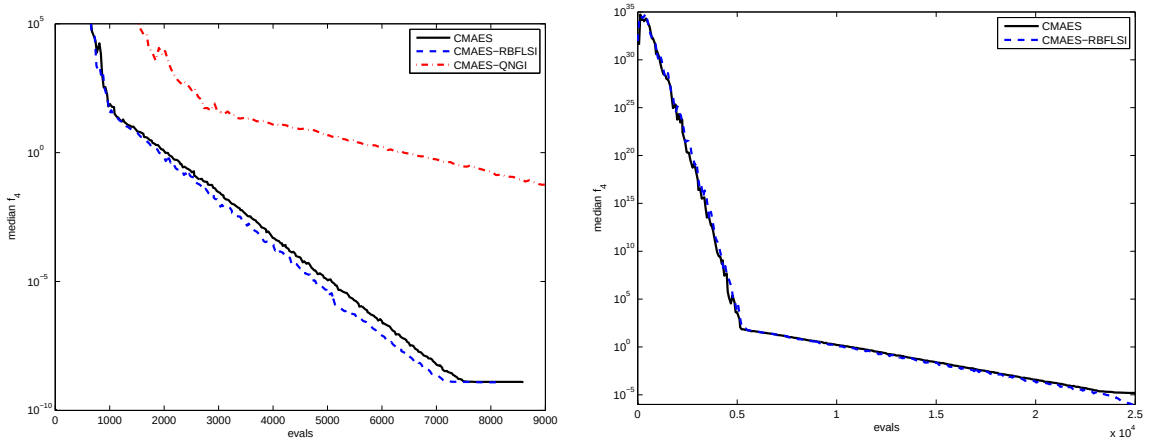
CMAES. This does not seem to be a case of simple error in implementing their strategy as we have been able to successfully implement and test it in the context of a standard evolution strategy; results not shown here. Rather the symmetrization of the population at each generation seems to interfere with the covariance matrix adaptation and/or step size adaptation algorithm in CMAES. It may be possible to develop a new version of CMAES which can accomodate the symmetrized population used in CMAES-QNGI, but that is beyond the scope of the current paper. The RBF local search individual approach does not have this difficulty since it does not modify the existing population of the evolution strategy other than simply appending an extra individual to the population.

A second observation is that for simple functions which are sufficiently smooth near their minima, such as the Schewefel 1.2 quadratic (f_1; Figure 3) and Rosenbrock (f_3; Figure 5) functions the CMAES-RBFLSI algorithm requires significantly fewer function



(a) (b)

Figure 5. Convergence graphs for, f_3, the generalized Rosenbrock function in 10 (a) and 30 (b) dimensions. It has two minima and is smooth, but has a long, flat and narrow valley that makes optimization slow for most algorithms. CMAES clearly benefits from local search as the minimum is approached.

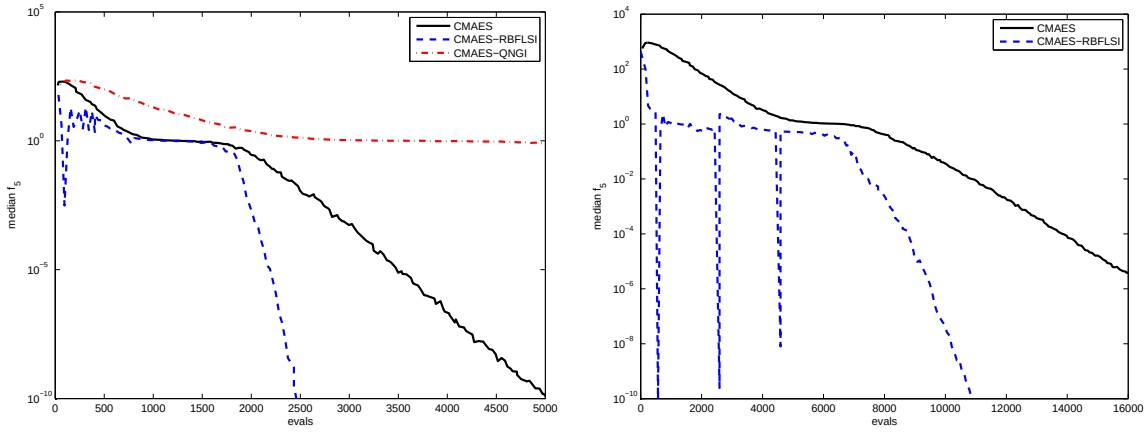


(a) (b)

Figure 6. Convergence graphs for, f_4, the Schwefel 1.5 function in 10 (a) and 30 (b) dimensions. It is unimodal with the minimum at $x = 0$, but is not differentiable if any $x_i = 0, i = 1, \ldots, n$.

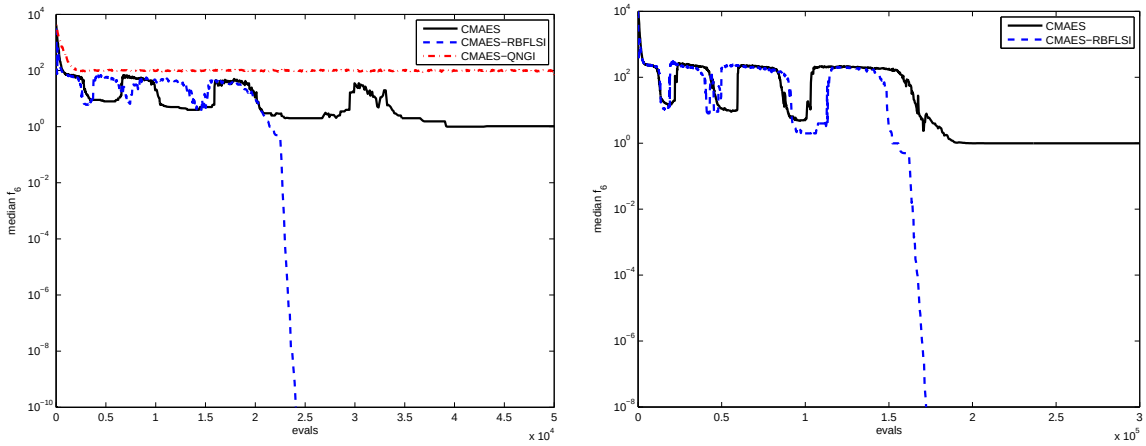
evaluations than standard CMAES. However, when the function is not differentiable at the minima the speedup due to the proposed algorithm is quite small. The latter is evident in the cone functon (f_2; Figure 4), Schwefel 1.5 function(f_4; Figure 6), and even the multimodal Ackley function (f_7; Figure 9).

The hybrid algorithm performs well for multimodal functions as well. For instance, in 10 and 30 dimensions there are significant reductions in the number of function evaluations for the Griewank function (f_5; Figure 7). The Griewank function is a relatively easy function for CMAES to optimize, but the CMAES-RBFLSI is able converge more quickly in the vicinity of the smooth global minimum. In fact, in many of the trials, the radial basis function local search individual is very close to the global minimum very early in the run, giving rise to the dips in the convergence curves for CMAES-RBFLSI in Figure



(a) (b)

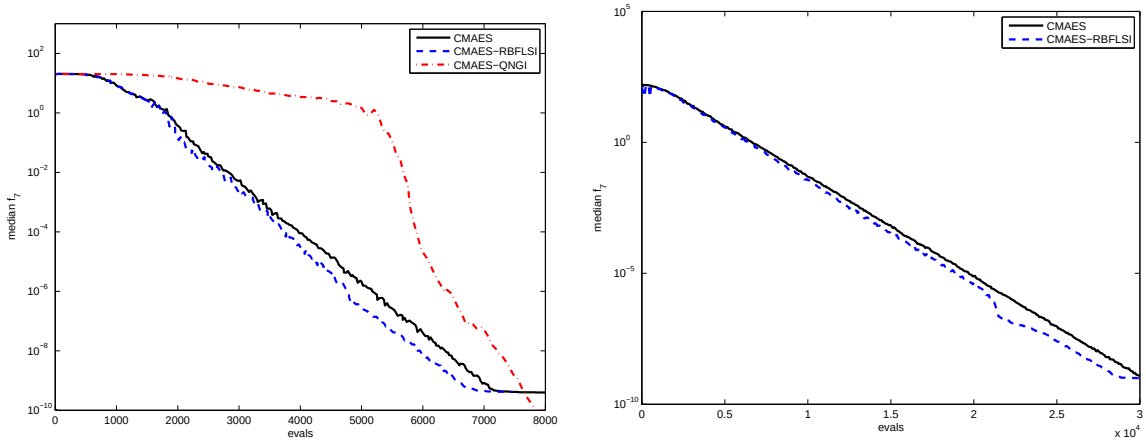
Figure 7. Convergence graphs for, f_5, the Griewank function in 10 (a) and 30 (b) dimensions. It is multimodal and smooth. While the Griewank function is not particularly challenging for CMAES, it nevertheless is greatly accelerated by inclusion of the local search individual. The radial basis function search individual often nearly finds the global optimum early in the optimization run giving rise to the dips in these convergence graphs.



(a) (b)

Figure 8. Convergence graphs for, f_6, the Rastrigin function in 10 (a) and 30 (b) dimensions. It is multimodal and smooth. For this function restarts and population doubling are used; see the text. This function is difficult for CMAES which rarely finds the global minimum, but CMAES-RBFLSI is much more reliable.

7. The Rastrign function (f_6; Figure 8) is more interesting as the population size needs to be quite large to locate the global minimum. For this function a population doubling restart scheme (Auger and Hansen 2005) was used in which the algorithm was restarted iteratively with population size $\lambda_k = 2^k\lambda, k = 1, 2, 3, 4$, and $\mu = \lambda_k/2$. In 10 dimensions CMAES locates the global minimum in only 5 of the 30 trials, while in 30 dimensions only 4 of the 30 trials. While with CMAES-RBFLSI, the global minimum is located in 22 or 30 trials in 10 dimensions and in 26 of 30 trials in 30 dimensions. Evidently, the proposed algorithm is able to more quickly converge to local minima in each restart and is thus able to use the computational budget more efficiently to find the global minimum.



(a) (b)

Figure 9. Convergence graphs for, f_7, the Ackley function in 10 (a) and 30 (b) dimensions. It is multimodal and not differentiable at the global minimum. For this function restarts and population doubling are used; see the text.

5. Application to calibration of a watershed model.

To demonstrate the utility of the CMAES-RBFLSI algorithm in a practical setting we used the algorithm to calibrate HYMOD, a five-parameter conceptual rainfall-runoff model (see Figure 11), introduced by Boyle (2000). In short, given time series of daily precipitation (P) and evapotranspiration (ET) data the objective is to tune the parameters so that the least squares error between the model predicted stream flow time series and the observed stream flow time series is minimized. Such problems are usually characterized by multiple minima, sometimes unidentifiable parameters and even discontinuities in the objective function (Duan *et al.* 1992).

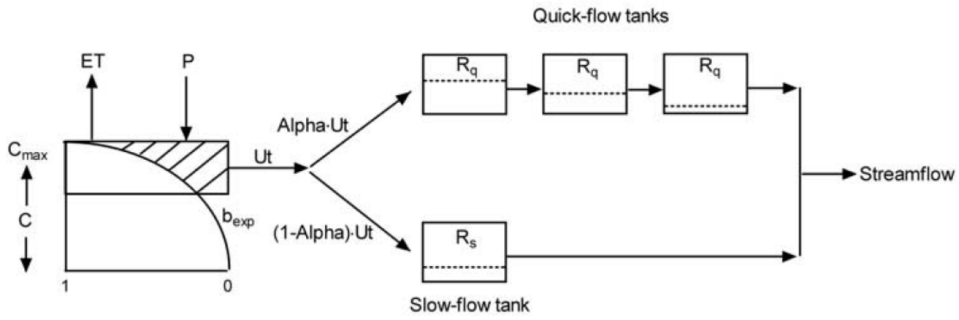


Figure 10. Schematic representation of the HYMOD model; from (Vrugt *et al.* 2003)

In attempts to parsimoniously represent the salient features of the precipitation-runoff process in a watershed system, the HYMOD model model structure (Moore 1985) is characterized by two series of linear reservoirs; in particular, three identical quick and a single slow response reservoir. The five parameters to be calibrated for the model stream flow to match the observed stream flow data are: the maximum storage capacity of the catchement, $C_{\max}$; the degree of spatial variability of the soil moisture capacity, $b_{\exp}$; the factor distributing flow between the two series of reservoirs, Alpha; and the residence

time of the linear quick and slow reservoirs, R_q and R_s, respectively. The hydrologic model parameters are inferred by adjusting their values until an acceptable level of agreement is achieved between a set of historical observations of the real world system that the model represents and their simulated counterparts. In this case, the objective function is simply the sum of the squared differences between the observed and simulated daily stream flows.

Three years, October 1, 1948, to September 30, 1951, of daily hydrologic data from the Leaf River watershed were used for model calibration. This humid 1944 km^2 watershed is located north of Collins, Mississippi. The data, obtained from the National Weather Service Hydrology Labratory, consist of mean areal precipitation (mm/d), potential evapotranspiration (mm/d), and stream flow (m^3/s).

Table 3. Uncertainty ranges of HYMOD model parameters

	Minimum	Maximum	Unit
$C_{\max}$	1.000	500.000	mm
$b_{\exp}$	0.100	2.000	
Alpha	0.100	0.990	
R_s	0.000	0.300	day
R_q	0.000	0.990	day

The CMAES and CMAES-RBFLSI algorithms are each applied to the optimization or calibration of the HYMOD model for 30 trials. The initial ranges of the parameter values are shown in Table 3; similar ranges of parameter values are used in model calibration study in (Vrugt *et al.* 2003). As for the mathematical test functions discussed above, the algorithms are initialized with the same uniform initial distributions. We set $\lambda = 10$ and $\mu = 5$. Each algorithm stops when TolFun$= 5e-4$, as described previously. For CMAES-RBFLSI the number of nearest neighbors used for the local RBF is $k = \lceil 1.5(n+1)(n+2)/2 \rceil = 23$ chosen from the last $N = 2k = 46$ evaluated points.

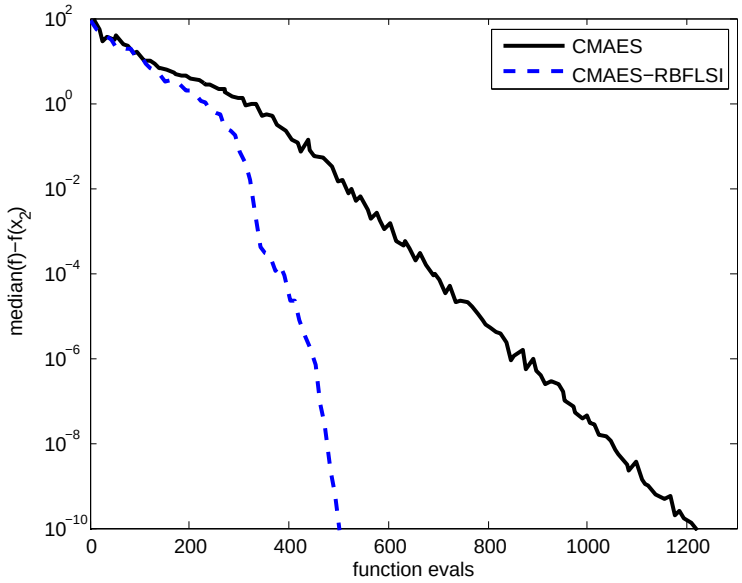


Figure 11. Convergence of CMAES and CMAES-RBFLSI for the HYMOD model.

There are many local minima, but CMAES and CMAES-RBFLSI nearly always converge to one of two minima $\mathbf{x}_1 = (157.0796, 0.5440, 0.2376, 0.2624, 0.8178)$ where

$f_{\text{HYMOD}}(\mathbf{x}_1) = 128.5346$ or $\mathbf{x}_2 = (146.9868, 0.7165, 0.2416, 0.2619, 0.8313)$ where $f_{\text{HYMOD}}(\mathbf{x}_2) = 128.6374$. The global minimum appears to be at $\mathbf{x}_1$ but CMAES converges to $\mathbf{x_2}$ in 28 of 30 trials, while CMAES-RBFLSI converges to the same minimum in 27 of 30 trials. Evidently, the basin of attraction for the global minimum, $\mathbf{x}_1$, is quite small as both algorithms have trouble finding it. The accelerated convergence of CMAES-RBFLSI to the local minimum of the HYMOD model is demonstrated in Figure 11. For each trial the best function value is saved at each generation. The median function value over the thirty trials minus the value at the local minimum , $f(\mathbf{x_2})$, is plotted versus the number of function evaluations. As Fiure 11 demonstrates, the increase in convergence speed is quite dramatic: CMAES-RBFGI typically converges with fewer than half of the objective function evaluations. Though neither algorithm reliably locates the global minimum, both algorithms give good approximations to the global minimum that produce adequate approximations to the daily stream flows. To locate the global minimum reliably, a restart strategy could be used as with the Rastrigin function above. The RBFLSI method would still accelerate the convergence significantly.

6. Conclusions

The local search individual hybridization approach for evolution strategies has been shown to be effective for significantly accelerating the convergence of the covariance matrix adaptation evolution strategy for functions which exhibit sufficient smoothness near the minimum. Likewise, it also works with the standard evolution strategy, though the results are not shown here.

To develop a hybrid evolution strategy using local RBF approximation, as we have done here, requires very little modification of the actual evolution strategy. The evolving population itself is not modified, but the additional local search individual is added at each generation. In the approach of Tahk *et al.* (2009) the population is chosen symmetrically at each generation and, as seen here, this can interfere with the convergence of CMAES.

Another advantage to this approach is that no minimum population size is required. In (Tahk *et al.* 2007, 2009) the population size must be at least twice the dimension of the search space to estimate the gradient vector. The downside of RBFGI approach is that it is expensive to form the coefficient matrix in Eq. (4). Moreover, the size of that system scales as $O(n^2)$, so its solution by a direct method requires $O(n^6)$ operations per generation. Thus, there is a trade-off between the computational complexity of the RBFLSI method and the gain due to fewer function evaluations. For expensive objective functions the cost of adding a local search individual propagated by local radial basis function approximation is expected to be incidental and the increase in speed can be enormous. As a result, the RBFLSI approach should be incorporated into evolution strategies for expensive functions as it can sometimes greatly increase converge speed and reliability with little downside.

In a future work we will consider a local search based on a trust region search along the Newton update direction on the local radial basis function approximation, as in (Wild *et al.* 2008) to locate the local search individual. The integration of a more robust local search may improve the local convergence properties of this hybridization approach.

Acknowledgements

The first author gratefully acknowledges the support of ARO Grant W911NF-06-0306. Permission was granted by the Chief of Engineers to publish this information.

References

Abbas, H., Bagirov, A., and Zhang, J., 2003. The discrete gradient evolutionary strategy for global optimization. *In*: *Evolutionary Computation, CEC '03*, 435–442.

Auger, A. and Hansen, N., 2005. A restart CMA evolution strategy with increasing population size. *In*: *Proc. IEEE Congr. Evol. Comput.*, Vol. 1, Edinburgh, U.K.., 1777–1784.

Baggett, J. and Skahill, B., 2010. Hybrid Optimization using Evolutionary and Approximate Gradient Search for Expensive Functions. *In*: *Proc. 2nd Int. Conf. Eng. Opt.*

Boyle, D., 2000. Multicriteria calibration of hydrological models. Thesis (PhD). Univ. Arizona.

Duan, Q., Gupta, V., and Sorooshian, S., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.*, 28, 1015–1031.

Fornberg, B., Flyer, N., and Russell, J., 2009. Comparisons between pseuospectral and radial basis function derivative approximations. *IMA Journal of Numerical Analysis*, 30 (1), 149–172.

Hansen, N., 2004. *Source Code for CMA-ES* [online]. INRIA, Orsay, France: N. Hansen. Available from: `http://www.lri.fr/~hansen/cmaes_inmatlab.html`, [Accessed July 2007].

Hansen, N., 2009. Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed. *In*: *Workshop Proc. GECCO Genetic Evol. Comp. Conf.*, Montreal, Canada., 2389–2395.

Hansen, N., 2010. *The CMA Evolution Strategy: A tutorial* [online]. INRIA, Orsay, France: N. Hansen. Available from: `http://www.lri.fr/~hansen/cmatutorial.pdf` [Accessed October, 2010].

Hansen, N. and Kern, S., 2004. Evaluating the CMA strategy on multimodal test functions. *In*: X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervos, J. Bullinaria, J. Rowe, P. Tino, A. Kaban and H.P. Schwefel, eds. *Parallel Problem Solving from Nature – PPSN VIII.*, Vol. 3242 of *Lecture Notes in Computer Science* Springer Verlag, 282–291.

Hansen, N. and Ostermeier, A., 1996. Completely arbitrary normal mutation distributions in evolution strategies. *In*: *Proc. IEEE Cong. Evol. Comput.*, Nagoya, Japan., 312–317.

Kern, S., Hansen, N., and Koumoutsakos, P., 2004. Local Meta-Models for Optimization Using Evolution Strategies. *In*: T. Runarsson, H.G. Beyer, E. Burke, J. Merelo-Guervs, L. Whitley and X. Yao, eds. *Parallel Problem Solving from Nature – PPSN IX.*, Vol. 4192 of *Lecture Notes in Computer Science* Springer Verlag, 282–291.

Min, B.M., *et al.*, 2007. Autopilot Design using Hybrid PSO-SQP Algorithm. *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques.*, Vol. 2 of *Communications in Computer and Information Science* Springer Verlag, 596–604.

Moore, R., 1985. The probability-distributed principle and runoff production at point and basin scales. *Hydrol. Sci. J.*, 30 (2), 273–297.

Peters, F., Barra, L., and Lemaoge, A., 2010. Application of a Hybrid Optimization Method for Identification of Steel Reinforcement in Concrete by Electrical

Impedance Tomography. *In*: *Proc. 2nd Int. Conf. Eng. Opt.*

Powell, M., 1992. The theory of radial basis function approximation in 1990. *In*: W. Light, ed. *Wavelets, Subdivision Algorithms and Radial Basis Functions.*, Vol. 2 of *Advances in Numerical Analysis* Oxford Univ. Press, 105–210.

Regis, R. and Shoemaker, C., 2004. Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions. *IEEE Trans. Evol. Comput.*, 8 (5), 490–505.

Renders, J.M. and Flasse, S., 1996. Hybrid Methods using Genetic Algorithms for Global Optimization. *IEEE Trans. Sys. Man. Cyb.*, 26 (2), 243–258.

Tahk, M.J., *et al.*, 2009. Hessian approximation algorithms for hybrid optimization methods. *Engineering Optimization*, 41 (7), 609–633.

Tahk, M.J., Woo, H., and Park, M., 2007. A hybrid optimization method of evolutionary and gradient search. *Engineering Optimization*, 39 (1), 87–104.

Vrugt, J., *et al.*, 2003. A shuffled complex evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resources Research*, 39.

Wild, S.M., Regis, R.G., and Shoemaker, C.A., 2008. ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM J. on Scientific Computing*, 30 (6), 3197–3219.

Woo, H., Kwon, H., and Tahk, M.J., 2004. A hybrid method of evolutionary algorithms and gradient search. *In*: S. Mukhopadhyay and G.S. Gupta, eds. *Proc. 2nd Int. Conf. Autonomous Robots and Agents*, Palmerston North, New Zealand.

Appendix C

More efficient Bayesian-based optimization and uncertainty assesment of hydrologic model parameters

Brian E. Skahill and Jeff S. Baggett

Fiscal Year 2010 Work Unit Report for Arid Regions Program

September, 2010.

More efficient Bayesian-based optimization and uncertainty assessment of hydrologic model parameters

Brian E. Skahill[1*] and Jeffrey S. Baggett[2]

[1]U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory, Hydrologic Systems Branch, Vicksburg, MS, USA

[2]University of Wisconsin at La Crosse, La Crosse, WI, USA

September 30, 2010

Introduction

Hydrologic models, regardless of their type (e.g., empirical, physics-based), often contain parameters that cannot be measured directly either because they have no physical basis, it would be impractical, or due to an incompatibility of scales, among other possible reasons. Hence, hydrologic model parameters are inferred by adjusting their values until an acceptable level of agreement is achieved between a set of historical observations of the real world system that the model represents and their simulated counterparts. While manual model calibration is certainly one approach to the problem, it is subjective, labor-intensive, and may also suffer from a lack of consistency and/or repeatability, among others. Moreover, it is difficult to imagine how even an experienced modeler would necessarily manage, in a manual calibration context, the large number of estimable parameters associated with present day practice driven complex hydrologic model deployments. Fortunately, the computer-based calibration of hydrologic models (which, in contrast with the manual approach to model calibration is more objective, repeatable, and better capitalizes on the computational capacity of the modern computer) is an active area of research and development (see; for example, Baggett and Skahill, 2010a; Baggett and Skahill, 2010b; Skahill et al., 2009, Skahill and Doherty, 2006; Doherty and Skahill, 2006, and references cited therein) which has resulted in numerous automatic

*Corresponding author. Email: Brian.E.Skahill@usace.army.mil

calibration methods that are readily available (see Mattot et al., 2009 and references cited therein) for the modern day hydrologic modeler to employ. And the knowledge gained by their application and development has provided the hydrologic modeling community with a better understanding of some of the complications associated with calibrating hydrologic models; viz., among others, the existence of multiple local optima, non-smooth objective function surfaces, and long valleys in parameter space that are a result of excessive parameter correlation or insensitivity (Gupta et al., 2003; Duan et al., 1992).

As mentioned, hydrologic models are typically calibrated by adjusting parameters encapsulated in the simulator until there is an acceptable level of agreement between a set of historical data and their model simulated counterparts. The parameters obtained via calibration are often then used by the model to predict system behavior for one or more pre-defined scenarios of interest to different groups whose life or livelihood is rooted in the local model study area. Regardless of the calibration method employed and the type (e.g., empirical or physics-based) of model used some if not all of the parameter values obtained through the calibration process possess a degree of quantifiable uncertainty because the observed data contain measurement errors and also because the model never perfectly represents the watershed system or exactly fits the observation data. And where model parameters are uncertain so too are model predictions. In particular, quantifying uncertainty supports, among others, the following hydrologic modeling activities (Schoups and Vrugt, 2010; Schoups et al., 2010):

1. model comparison and selection,
2. identification of the best water management strategies that reflect the likelihood of outcomes,
3. data collection aimed at improving hydrologic predictions and water management, and
4. regionalization and extrapolation of hydrologic parameters to ungauged basins.

For example, regarding item 4 above, to quote Vrugt et al. (2003a), “If we want to be able to regionalize or relate model parameters to easily measurable land or soil-surface characteristics,

a prerequisite is that the parameters be unique, preferably having a small variance. From this perspective, it is necessary to infer the parameter uncertainty resulting from calibration studies."

Model uncertainty, characterized by the model covariance matrix calculated using the model Jacobian evaluated at the best estimate for the model, can be quantified by employing a traditional linear analysis. However, this approach is local which does not dovetail well with the understanding that for hydrologic models there may exist many effectively equally acceptable models; i.e., it is difficult to identify a unique best estimate; and moreover, that the set of good models may very well not necessarily even be a closed and bounded set interior to feasible parameter space. In addition, it relies on a linearity assumption that is often violated in hydrologic modeling practice.

Bayesian-based approaches to model calibration, wherein a prior distribution for the model is proposed, and the vector of adjustable model parameters is treated as a random variable with a target probability distribution that is conditioned with observed data, are a formal means to obtain a realistic and reliable estimate of model uncertainty. In particular, Markov Chain Monte Carlo (MCMC) simulation, which is by far more efficient than other Monte Carlo methods, is used for inference, search, and optimization with hydrologic models (Harmon and Challenor, 1997; Kuczera and Parent, 1998; Campbell et al., 1999; Campbell and Bates, 2001; Makowski et al., 2002; Qian et al., 2003; Kanso et al., 2003; Vrugt et al., 2003a; Vrugt et al., 2003b; Vrugt et al., 2008). With MCMC we are interested in a target probability distribution, and its key elements include exploration of this distribution by way of some sort of random walk or diffusion process that must be initialized in an arbitrary way because we don't know a priori where good places necessarily are in parameter space. The random walk is directed by Markov chain simulation wherein the next step only depends on the previous step, and eventually after a burn in period the target distribution is identified.

An obvious advantage of the MCMC method is that no assumptions of model linearity, or even of differentiability of model outputs with respect to parameter values, are required for its implementation; hence it is extremely robust. However, this robustness comes at a cost, this

being the high number of model runs required for its implementation. The choice of the proposal distribution, which expresses prior information about the model, can greatly affect the efficiency of an MCMC sampler. A poorly chosen proposal distribution will result in slow convergence to the target distribution. Unfortunately, for complex hydrologic models there is very little a priori knowledge of the high probability density region within parameter space. Hence, with hydrologic models an uninformative prior; wherein all parameters have equal likelihood, may often be the best we can do. Clearly, for hydrologic modeling, there is a need to design efficient MCMC samplers, and this observation was the motivation for the development of the Shuffled Complex Evolution Metropolis algorithm (SCEM-UA), an effective and efficient adaptive MCMC sampler which tunes the proposal distribution during the evolution to the stationary posterior target distribution (Vrugt et al., 2003a).

The SCEM-UA algorithm is a modification to the SCE-UA global optimization algorithm (Duan et al., 1992; Duan et al., 1993). There are two primary modifications both which prevent SCEM-UA from collapsing into the region of a single best parameter set. The first modification involves replacement of the downhill simplex method with the Metropolis-annealing scheme (Metropolis et al., 1953). The second modification is that SCEM-UA does not further subdivide the complex into subcomplexes during the generation of candidate points and it uses a different replacement procedure. In presenting the SCEM-UA algorithm, Vrugt et al. (2003a) noted that their principal focus was algorithm efficiency; viz., the number of simulations needed to converge to the stationary posterior probability distribution. They compared the traditional Metropolis-Hastings sampler (Metropolis et al., 1953; Hastings, 1970) with SCEM-UA to infer the posterior distribution of five parameters associated with a conceptual rainfall-runoff model. It took SCEM-UA approximately 4,000 simulations to converge to the stationary posterior distribution, based on evaluation of the Gelman and Rubin convergence diagnostic (Gelman and Rubin, 1992); whereas, even after 30,000 simulations the Metropolis-Hastings algorithm had not converged to the target distribution when applying the same convergence criteria.

An important consideration in assessing the performance of model calibration software is that of run time. Model calibration software, no matter what its algorithmic basis, must run the hydrologic model to be calibrated many times in the course of minimizing the objective function that is used to characterize model-to-measurement misfit. Minimizing the number of hydrologic model runs required during the calibration process is nearly always important, but particularly when the objective function landscape contains multiple local minima or hydrologic model run times are high. Minimizing the number of required model runs is the primary factor driving the research and development to be discussed herein, such that the resulting optimization and uncertainty tool is compatible with the computationally expensive physics-based models that are becoming more commonly used within the practice community.

Recent research and development activity directed at improving the efficiency of native computer-based model calibration algorithms includes the work of Skahill et al. (2009) and Baggett and Skahill (2010a,b), among others. Skahill et al. (2009) developed an accelerated derivative-based local search algorithm and based on three separate modeling applications demonstrated efficiency gains anywhere from 36-84% in comparison with the native algorithm. Baggett and Skahill (2010a,b) reported on an efficiency enhancement to the state-of-the-art covariance matrix adaption evolution strategy (CMAES) (Hansen, 2006) for global parameter identification of difficult problems with noise or other features that make derivatives estimation difficult. The increase in convergence speed was quite dramatic for their modified CMAES algorithm, which uses a local radial basis function approximation to the objective function to compute approximate first and second derivatives to the objective function surface to propagate a gradient individual alongside the evolving population for possible selection each generation. Based on a summary of thirty trials, it converged with fewer than half of the objective function evaluations required by CMAES when applied to calibrate a hydrologic model.

The primary objective of the research and development encapsulated in this report was to improve upon the already existing documented efficiency of an existing state-of-the-art Bayesian model uncertainty analysis method (Vrugt et al., 2003a). The principal approach that

was employed to achieve the stated objective was to simultaneously and adaptively construct an approximation to the objective function.

Background

The hydrologic model *f* can be written as

$$\hat{\mathbf{y}} = f(\boldsymbol{\theta}; \mathbf{x}) + \boldsymbol{e} \quad (1)$$

where $\hat{\mathbf{y}}$, $\mathbf{x}$, $\boldsymbol{\theta}$, and $\boldsymbol{e}$ represent, respectively, the vector of model outputs, structural aspects of the model, as well as its input dataset, model parameters that are adjustable through the calibration process, and the vector of statistically independent error terms with zero expectation and constant variance σ^2. Given the vector of observations $\mathbf{y}$, the vector of residuals is given by

$$\boldsymbol{e}(\boldsymbol{\theta}) = \hat{\mathbf{y}} - \mathbf{y} \quad (2)$$

Bayesian statistics treats the model parameters $\boldsymbol{\theta}$ as probabilistic variables having a joint posterior probability density function (pdf), $p(\boldsymbol{\theta}|\mathbf{y})$. The posterior pdf is a probabilistic statement about the parameters $\boldsymbol{\theta}$ conditioned on the observed data $\mathbf{y}$. At the core of Bayesian inference is Bayes' rule, which is given by

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto L(\mathbf{y}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}) \quad (3)$$

where $p()$ indicates probability, $p(\boldsymbol{\theta}|\mathbf{y})$ is the posterior probability distribution of the parameters $\boldsymbol{\theta}$, $L(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function, and $p(\boldsymbol{\theta})$ is the prior probability density function. The prior pdf, $p(\boldsymbol{\theta})$, represents information about $\boldsymbol{\theta}$ before any data are collected. A critical term in Bayes' rule is the likelihood term; likelihoods can only be calculated if an error model is available. Assuming that the residuals are mutually independent, Gaussian distributed, with constant variance, and further assuming a noninformative prior of the form $p(\boldsymbol{\theta}) \propto \sigma^{-1}$, Box and Tiao (1973) derived the following form for the posterior probability distribution of $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto [M(\boldsymbol{\theta})]^{-N(1+\gamma)/2} \quad (4)$$

where

$$M(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(e(\theta)\right)^{2/(1+\gamma)} \quad (5)$$

The Shuffled Complex Evolution Metropolis Function Approximation Algorithm

The goal of the Shuffled Complex Evolution Metropolis (SCEM-UA) algorithm, a Markov Chain Monte Carlo sampler developed by Vrugt et al. (2003a) which is a modified version of the original SCE-UA global optimization algorithm developed by Duan et al. (1992) was to efficiently (and by efficiency, we mean the number of forward model calls necessary to converge to the target posterior distribution) infer the posterior distribution of hydrologic model parameters. The SCEM-UA algorithm is not only able to effectively infer the posterior distribution of hydrologic model parameters but also the most likely parameters within this high probability density region. Function approximation methods have successfully been employed to improve upon the efficiency of native evolutionary strategies utilized for model calibration; for example, see Baggett and Skahill (2010a,b) and references cited therein. By interfacing function approximation methods with the native SCEM-UA algorithm, we further improve upon the already existing reported efficiencies of the SCEM-UA MCMC sampler. The new algorithm presented in full herein, entitled the Shuffled Complex Evolution Metropolis Function Approximation (SCEM-FA) algorithm, is given below and also illustrated in Figures 1 and 2. The SCEM-FA algorithm retains all of the elements of the original SCEM-UA algorithm. And we refer the interested reader to Vrugt et al. (2003a) for a thorough description and discussion of the original SCEM-UA algorithm. However, for purposes of completeness, we present the entire SCEM-FA algorithm. We will emphasize those parts which constitute the existing function approximation interface to the original SCEM-UA algorithm. For further clarity regarding the SCEM-FA and SCEM-UA algorithms, we make mention now of the fact that the function approximation interface methodology presented herein is not only possible with SCEM-UA, but

likely could also easily be adapted and employed with other MCMC samplers; for example, DREAM (Vrugt et al., 2008).

1. Generate sample. Sample s $\{\theta_1, \theta_2, ..., \theta_s\}$ points randomly from the prior distribution and compute the posterior density of each point $\{p(\theta^{(1)}|\mathbf{y}), p(\theta^{(2)}|\mathbf{y}), \ldots, p(\theta^{(s)}|\mathbf{y})\}$ using equation (4).
2. Rank points. Sort the s points in order of decreasing posterior density and store them in array D[1:s, 1:n + 1] where n is the number of parameters, so that the first row of D represents the point with the highest posterior density.
3. Build function approximation. Train a locally weighted projection regression function (LWPR) approximation (Vijayakumar et al., 2005) using the s points randomly sampled from the prior distribution. If the sample is small, then present the sample to LWPR multiple times in random order.
4. Initialize parallel sequences. Initialize the starting points of the parallel sequences, S^1, S^2, …, S^q, such that S^k is D[k, 1:n + 1], where k = 1, 2, ..., q.
5. Partition into complexes. Partition the s points of D into q complexes C^1, C^2, …, C^q, each containing m points, such that the first complex contains every q(j-1)+1 ranked point, the second complex contains every q(j-1)+2 ranked point of D, and so on, where j = 1, 2, …, m.
6. Evolve each sequence. Evolve each of the parallel sequences according to the Sequence Evolution Metropolis Function Approximation (SEM-FA) algorithm outlined below.
7. Adjust SEM-FA input value r. Based on the monitored acceptance rate in SEM-FA, and predefined input values for an acceptance rate threshold for SEM-FA, and the occurrence frequency for SEM-FA input parameter adjustment, update the SEM-FA input, r, a number in the interval (0,1) which effectively dials in or out the employment of function approximation in SEM-FA.
8. Shuffle complexes. Unpack all complexes C back into D, rank the points in order of decreasing posterior density, and reshuffle the s points into q complexes according to the procedure specified in step 5.

9. Check convergence. Check the Gelman and Rubin convergence statistic (Gelman and Rubin, 1992). If convergence criteria are satisfied, stop; otherwise, return to step 6.

Items 3, 6, and 7 above are specific to the SCEM-FA algorithm while the other elements are a restatement of the native SCEM-UA algorithm originally presented in Vrugt et al. (2003a). The first modification, listed in item 3 above, uses the initial random sample to build a function approximation model which is later used in SEM-FA as a surrogate for the objective function. While locally weighted progression regression (Vijayakumar et al., 2005) was the function approximation method used for the current modifications to the SCEM-UA algorithm documented in this report, alternative function approximation methods, such as radial basis functions (Powell, 1992), could also have been used. Item 6 above refers to the SEM-FA algorithm which is presented and discussed below while item 7 above refers to the current method that is employed to regulate the degree to which the function approximation model is utilized in SEM-FA. Both items 6 and 7 will be discussed further below.

As Vrugt et al. (2003a) mention, one of the essential elements of the SCEM-UA algorithm is the Sequence Evolution Metropolis (SEM) algorithm, wherein new candidate points are produced in each of the parallel sequences S^k and the Metropolis algorithm is used to test whether or not the candidate point should be added to the current sequence. As part of the overall effort to further improve upon the already existing reported efficiencies of the SCEM-UA MCMC sampler, the SEM algorithm was adapted to include a function approximation model which is used as a surrogate for the objective function. It is named SEM-FA for Sequence Evolution Metropolis Function Approximation and it is presented below and also in Figure 2. As with the previously mentioned comparison of the SCEM-UA and SCEM-FA algorithms, the SEM-FA algorithm retains all of the elements of the original SEM algorithm. And we refer the interested reader to Vrugt et al. (2003a) for a thorough description and discussion of the original SEM algorithm. However, for purposes of completeness, we present the entire SEM-FA algorithm. We will emphasize those parts which constitute the existing function approximation interface to the original SEM algorithm.

I. Compute the mean, μ^k, and covariance structure $\sum^k$ of the parameters of C^k. Sort the *m* points in complex C^k in order of decreasing posterior density and compute Γ^k, the ratio of the posterior density of first ("best") to the posterior density of the last ("worst") member of C^k.

II. Compute α^k, the ratio of the mean posterior density of the *m* points in C^k to the mean posterior density of the last *m* generated points in S^k.

III. If α^k is smaller than a predefined likelihood ratio, T, generate a candidate point, $\theta^{(t+1)}$, by using a multinormal distribution centered on the last draw, $\theta^{(t)}$, of the sequence, S^k, and covariance structure $c_n^2\sum^k$, where c_n is a predefined jumprate. Go to step V, otherwise, continue with step IV.

IV. Generate offspring, $\theta^{(t+1)}$, by using a multinormal distribution with mean μ^k and covariance structure $c_n^2\sum^k$, and go to step V.

V. If the random number from the interval (0,1), input from SCEM-FA, is less than r, then use the function approximation to compute the posterior dentisty, $p(\theta^{(t+1)}|\mathbf{y})$, of $\theta^{(t+1)}$ using equation (4). If the generated candidate point is outside the feasible parameter space, then set $p(\theta^{(t+1)}|\mathbf{y})$ to zero.

VI. If the random number from the interval (0,1), input from SCEM-FA, is greater than or equal to r, then perform a forward model call, compute the posterior dentisty, $p(\theta^{(t+1)}|\mathbf{y})$, of $\theta^{(t+1)}$ using equation (4), and update the LWPR function approximation with the new data point $\theta^{(t+1)}$. If the generated candidate point is outside the feasible parameter space, then set $p(\theta^{(t+1)}|\mathbf{y})$ to zero.

VII. Randomly sample a uniform label Z over the interval 0 to 1.

VIII. If the random number from the interval (0,1), input from SCEM-FA, is less than r, then go to step IX; otherwise, go to step XII.

IX. Compute the ratio $\Omega = p(\theta^{(t+1)}|\mathbf{y})/ p(\theta^{(t)}|\mathbf{y})$. If Z is smaller than or identical to Ω, then perform a forward model call, compute the posterior dentisty, $p(\theta^{(t+1)}|\mathbf{y})$, of $\theta^{(t+1)}$ using equation (4), and update the LWPR function approximation with the new data point $\theta^{(t+1)}$. If the generated candidate point is outside the feasible parameter space, then set $p(\theta^{(t+1)}|\mathbf{y})$ to zero.

X. However, if Z is larger than Ω, reject the candidate point and remain at the current position in the sequence, that is, $\theta^{(t+1)} = \theta^{(t)}$. Go to step XIII.

XI. Recompute the ratio $\Omega = p(\theta^{(t+1)}|\mathbf{y})/ p(\theta^{(t)}|\mathbf{y})$. If Z is smaller than or identical to Ω, then accept the new candidate point. However, if Z is larger than Ω, reject the candidate point and remain at the current position in the sequence, that is, $\theta^{(t+1)} = \theta^{(t)}$. Go to step XIII.

XII. Compute the ratio $\Omega = p(\theta^{(t+1)}|\mathbf{y})/ p(\theta^{(t)}|\mathbf{y})$. If Z is smaller than or identical to Ω, then accept the new candidate point. However, if Z is larger than Ω, reject the candidate point and remain at the current position in the sequence, that is, $\theta^{(t+1)} = \theta^{(t)}$.

XIII. Add the point $\theta^{(t+1)}$ to the sequence S^k.

XIV. If the candidate point is accepted, replace the best member of C^k with $\theta^{(t+1)}$, and go to step XV; otherwise replace the worst member (*m*) of C^k with $\theta^{(t+1)}$, provided that Γ^k is larger than the predefined likelihood ratio, T, and $p(\theta^{(t+1)}|\mathbf{y})$ is higher than the posterior density of the worst member of C^k.

XV. Repeat the steps I – XIII L times, where L is the number of evolution steps taken by each sequence before complexes are shuffled.

Items I – IV, VI, VII, and XII – XV (with the SCEM-FA input value r set to zero) are a restatement of the native SEM algorithm originally presented in Vrugt et al. (2003a). The SCEM-FA algorithm is equivalent to the original SCEM-UA algorithm when the SCEM-FA input parameter r is set to a value of zero. Items I – XV directly above list the existing modification to the original SEM algorithm.

The basic reasoning behind SEM-FA is that if the function approximation prediction, which is used as a surrogate for the objective function, suggests that the candidate point should be selected, by way of evaluation of the Metropolis algorithm criterion (Metropolis et al., 1953), then proceed ahead with a forward model call and re-evaluation of the Metropolis algorithm criterion to determine if in fact the candidate point is to be accepted or not. And if the Metropolis algorithm criterion computed using the function approximation prediction indicates

otherwise, then reject the candidate point. In effect, the function approximation prediction serves as a screening device in that forward model calls are only performed when it suggests that it would be beneficial. And the degree to which the filter is applied is based on a SCEM-FA input parameter, r, which is dynamically adjusted during SCEM-FA execution, and its comparison (see Figure 2) with a unique randomly sampled uniform label over the interval 0 to 1 that is passed to SEM-FA for the evolution of each sequence (see Figure 1).

If the SCEM-FA input value for r is greater than zero, then the function approximation adaptations described above and also shown in Figures 1 and 2 will be active. In this case, the value for r is reset to zero at the beginning of SCEM-FA execution and dynamic adjustment is subsequently based not only on a comparison of the candidate point acceptance rate within SEM-FA with a user specified acceptance rate threshold, but also the integer value for a persistence parameter which determines the frequency for updating the value for r. In particular, at present, if it is an opportunity to update r and the SEM-FA acceptance rate is less/greater than the user specified acceptance rate threshold, then decrease/increase the value for r by one-tenth. The minimum possible value for r is zero, and its maximum is equivalent to its specified input value. At present, an input value is specified for r. However, it could possibly be effectively removed as an input for SCEM-FA by altering the existing dynamic adjustment process to simply allow the value for r to vary between zero and one. Testing is needed to explore this potential opportunity. If it is not already clear to the reader, decreasing/increasing the value for r increases/decreases the number of forward model calls within SEM-FA.

Guidance for the proper selection of SCEM-UA algorithmic input parameter values can be found in Vrugt et al. (2003a). The SCEM-FA algorithm contains three additional parameters that at present need to be specified by the user. These are (1) the random number threshold, r, (2) the acceptance rate threshold, a, and (3) the parameter, p, an integer value which determines the update frequency for r. The increment/decrement value embedded in the dynamic adjustment process for r could also be viewed as a parameter that could possibly impact SCEM-FA performance. Further exploration in terms of how these parameters affect the reliability (i.e.,

the capacity to converge to the same posterior probability distribution as the native SCEM-UA algorithm) and efficiency of SCEM-FA is needed before any recommendations can be provided for default values. However, optimal acceptance rates for MCMC samplers range anywhere from 20 – 70% in the literature (Gallagher and Doherty, 2007).

Additional opportunities exist, of course, for further exploration in terms of their potential capacity to yield additional efficiency gains beyond those already achieved and documented below with the existing SCEM-FA implementation. These include, among others, (1) relaxing the current requirement to perform a forward model call every time the function approximation suggests that the candidate point is to be accepted, and (2) comparing the current function approximation model with an alternative model, such as radial basis functions (Powell, 1992). Both of these opportunities would be modest development efforts.

With respect to the first opportunity noted directly above, at present, SCEM-FA is biased conservatively in that we completely trust the function approximation prediction to reject candidate points; whereas, if the function approximation prediction suggests that the candidate point is to be accepted, then we go to additional measures to ensure that is in fact the case. One alternative would be to simply accept the candidate point when the function approximation prediction suggests it should; however, that approach may be too aggressive and impair the reliability of SCEM-FA. A relatively simple easily implementable approach would be to monitor the success rate of the function approximation prediction and use that as a basis for deciding whether to perform a forward model call after the function approximation prediction suggests the candidate point is to be accepted. The second opportunity mentioned directly above would be a fairly modest effort because early SCEM-FA development utilized a k-nearest neighbor cubic radial basis function (RBF) local function approximation model rather than locally weighted projection regression (LWPR).

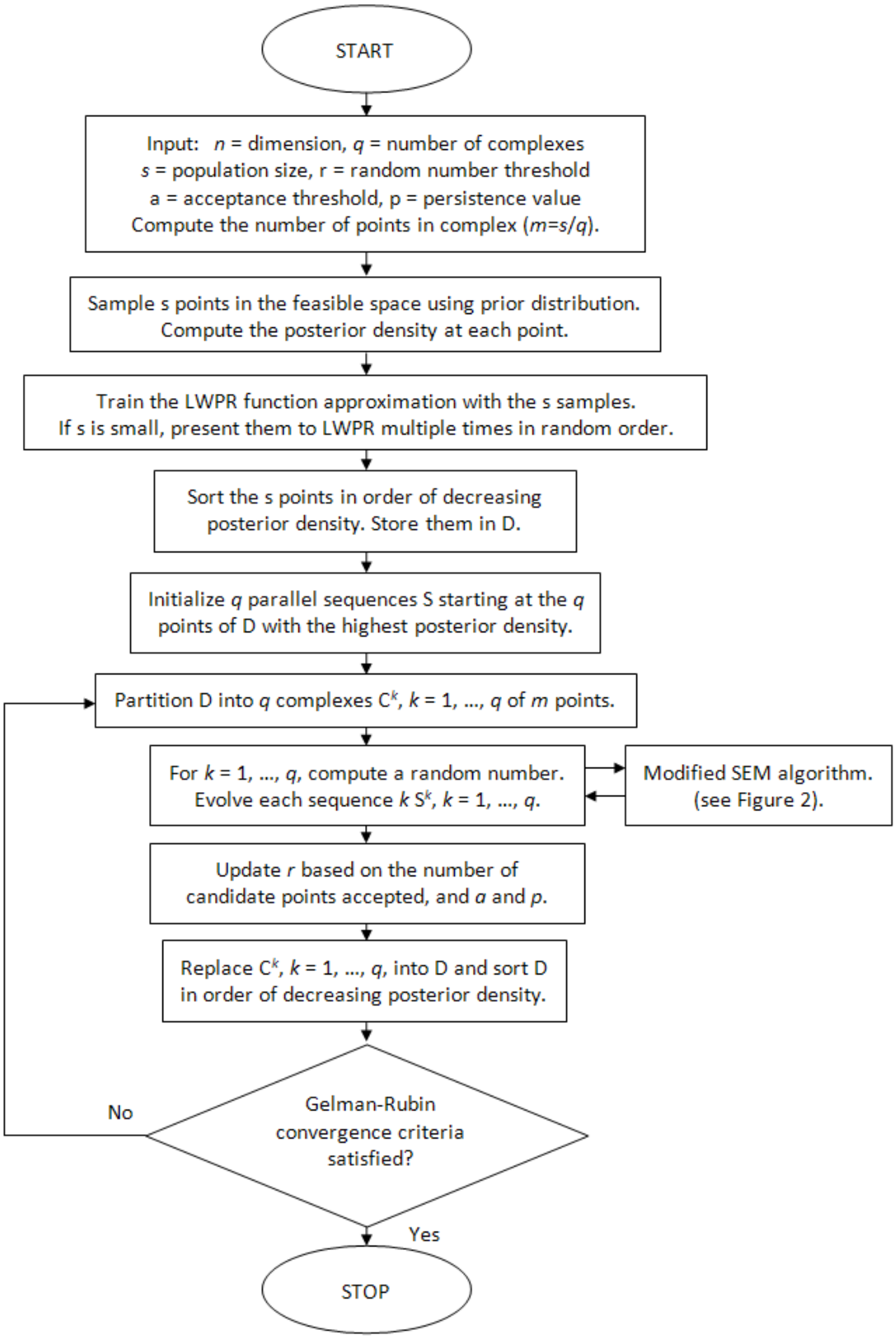


Figure 1. Flowchart of the SCEM-FA algorithm.

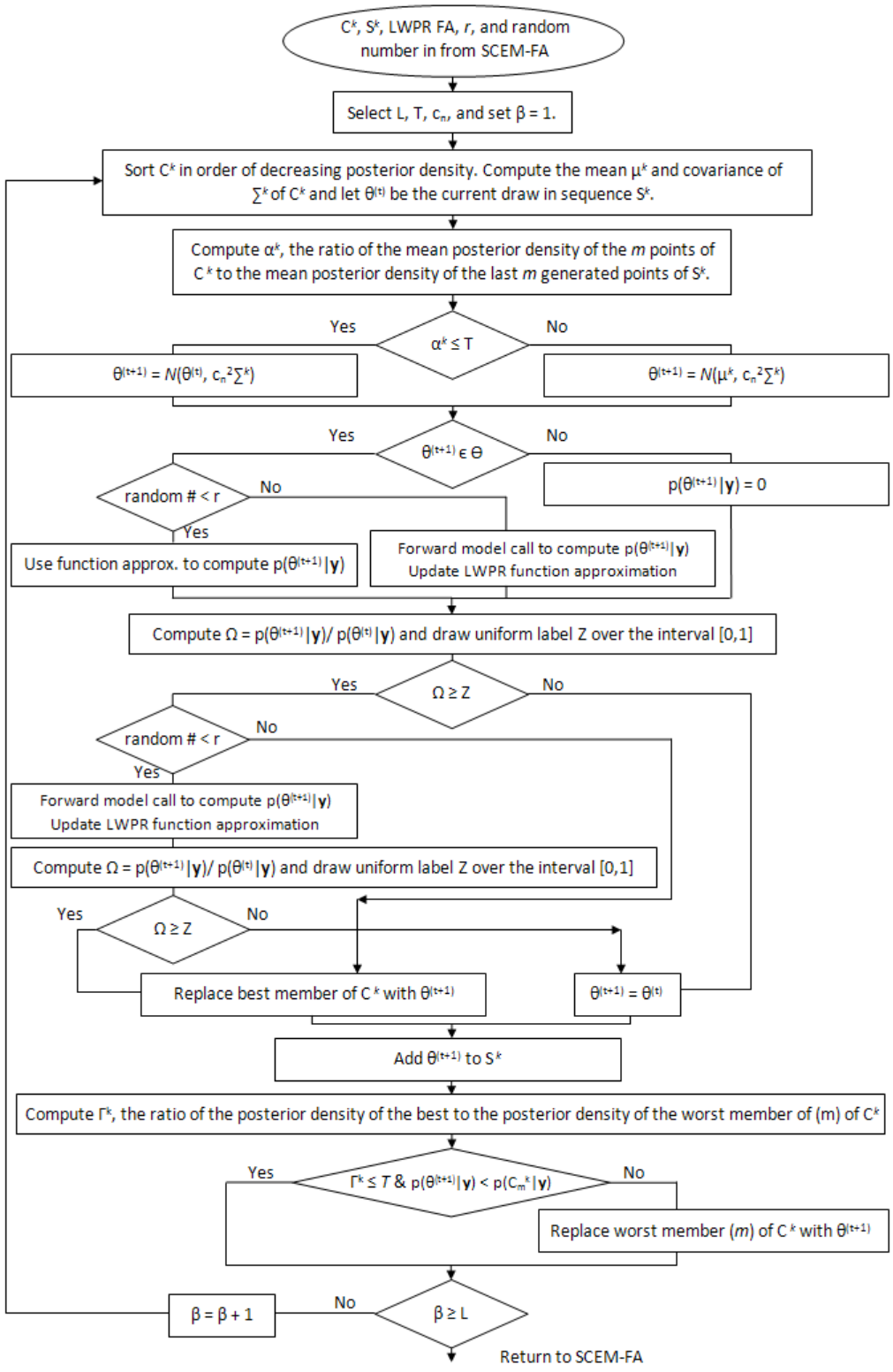


Figure 2. Flowchart of the SEM-FA algorithm employed in the SCEM-FA algorithm.

Case Study

To comprehensively demonstrate the efficiency gains that can be achieved with the SCEM-FA development efforts to date, all the while maintaining consistency with respect to convergence to the same target distribution, thirty unique instances of SCEM-UA and SCEM-FA were each employed to infer the posterior distribution of thirteen Sacramento soil moisture accounting (SAC-SMA) hydrologic model parameters using hydrologic data from the 1944 km^2 Leaf River watershed near Collins, MS. The SAC-SMA hydrologic model is used by the National Weather Service (NWS) for flood forecasting throughout the United States. While it has 16 parameters that need to be specified by the user, consistent with previous work (see Vrugt et al., 2003b and references cited therein), 13 were specified as adjustable. The prior uncertainty ranges of the specified adjustable SAC-SMA hydrologic model parameters are defined in Table 1. The reader is referred to (see Vrugt et al., 2003b and references cited therein) for comprehensive discussions regarding the SAC-SMA hydrologic model, the Leaf River watershed, and also its related hydrologic data (viz., mean areal precipitation (mm/day), potential evapotranspiration (mm/day), and streamflow (m3/s)) that was used to support the effective inference of the posterior distribution of the SAC-SMA adjustable model parameters and also the most likely parameters within this high probability density region.

Results associated with the thirty trials are summarized in Tables 2 – 8 and Figure 3. The results presented in Tables 2 – 5 are associated with an earlier version of SCEM-FA wherein the input parameter r was fixed and not dynamically adjusted as it is now, based on the candidate point acceptance rate, an acceptance rate threshold, and the persistence parameter, p, which dictates the update frequency for r. Examining the results in Table 2, we clearly see as one would expect, improved efficiency for SCEM-FA relative to SCEM-UA as the value for r increases. However, the improved efficiency that is obtained through more aggressive utilization of the function approximation prediction comes at the cost of decreased effectiveness in terms of convergence to the same posterior probability distribution as SCEM-UA, evidenced upon inspection of the lower order statistics for the objective function and SAC-SMA parameter values that are presented in Tables 3 – 5.

In attempts to balance efficiency with effectiveness, different heuristics for controlling the activation of the function approximation prediction within SCEM-FA were subsequently implemented, resulting in the existing SCEM-FA implementation documented in this report. Based on the thirty trials, the average number of forward model calls for SCEM-UA was 87,253; whereas, it was 68,642 with SCEM-FA, resulting in an approximate 21% reduction in total forward model calls. Comparing lower order statistics associated with the objective function and related parameter values obtained from samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA, as shown in Tables 6 – 8, it is clear that the existing SCEM-FA algorithm converged to the same target distribution as SCEM-UA. The results presented in Figure 3, marginal posterior probability distributions of the SAC-SMA model parameters based on 15,000 (500 for each of the 30 trials) samples generated with the SCEM-UA and SCEM-FA algorithms after convergence has been achieved with SCEM-UA and SCEM-FA, further confirm this observation. The results presented in Tables 6 – 8 and Figure 3 were obtained with SCEM-FA input parameters set to r = 0.9, a = 0.35, and p = 3.

Parameter	Minimum	Maximum	Unit
UZTWM	1	150	[mm]
UZFWM	1	150	[mm]
UZK	0.1	0.5	day^{-1}
PCTIM	0	0.1	[-]
ADIMP	0	0.4	[-]
ZPERC	1	250	[-]
REXP	1	5	[-]
LZTWM	1	500	[mm]
LZFSM	1	1000	[mm]
LZFPM	1	1000	[mm]
LZSK	0.01	0.25	day^{-1}
LZPK	0.0001	0.025	day^{-1}
PFREE	0	0.6	[-]

Table 1. Prior Uncertainty Ranges of the SAC-SMA Model Parameters.

Total Model Calls						
		SCEM-FA				
		Random Number Threshold in SEM-FA				
	SCEM-UA	0.1	0.3	0.5	0.7	0.9
Average	87253	79379	70602	66852	55158	50090
% reduction		9.0	19.1	23.4	36.8	42.6

Table 2. Summary of efficiency for an earlier version of SCEM-FA, relative to SCEM-UA, for fixed values of r. Results are based on thirty unique instances of the earlier version of SCEM-FA and also SCEM-UA.

Method	Objective Function Values	
	Average	Standard Deviation
SCEM-UA	13.31669413	0.02331727
SCEM-FA (r=0.1)	13.3073988	0.030142989
SCEM-FA (r=0.3)	13.35390347	0.21756169
SCEM-FA (r=0.5)	13.3382782	0.218278563
SCEM-FA (r=0.7)	13.42101667	0.325857754
SCEM-FA (r=0.9)	13.6533664	0.252643413

Table 3. Summary of objective function value lower order statistics for an earlier version of SCEM-FA, relative to SCEM-UA, for fixed values of r. Each individual result is based on thirty unique instances for the earlier version of SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

Method	Adjustable Model Parameters Average Values												
	UZTWM	UZFWM	UZK	PCTIM	ADIMP	ZPERC	REXP	LZTWM	LZFSM	LZFPM	LZSK	LZPK	PFREE
SCEM-UA	26.8812	29.5574	0.3811	0.0040	0.2722	203.3623	3.7797	250.7199	48.1212	105.5772	0.2364	0.0200	0.1502
SCEM-FA (r=0.1)	26.8277	29.2832	0.3798	0.0037	0.2762	206.1732	3.7819	249.4870	48.6453	105.7939	0.2367	0.0199	0.1472
SCEM-FA (r=0.3)	34.7161	35.9214	0.3805	0.0057	0.2845	204.0555	3.7810	248.0950	97.5675	136.9293	0.2387	0.0191	0.1571
SCEM-FA (r=0.5)	35.2474	36.3110	0.3848	0.0058	0.2847	209.6409	3.8505	249.8837	97.2791	136.5281	0.2402	0.0198	0.1629
SCEM-FA (r=0.7)	51.8347	45.8982	0.3906	0.0098	0.3013	204.0789	3.9138	239.9127	187.4767	199.7868	0.2422	0.0177	0.1851
SCEM-FA (r=0.9)	111.4248	66.9965	0.4062	0.0214	0.3526	185.1139	4.1644	200.3526	218.5091	212.8119	0.2413	0.0182	0.2908

Table 4. Summary of SAC-SMA average parameter values obtained from an earlier version of SCEM-FA, for fixed values of r, and also SCEM-UA. Each individual result is based on thirty unique instances for the earlier version of SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

Method	Adjustable Model Parameters Standard Deviation Values												
	UZTWM	UZFWM	UZK	PCTIM	ADIMP	ZPERC	REXP	LZTWM	LZFSM	LZFPM	LZSK	LZPK	PFREE
SCEM-UA	4.2085	2.6528	0.0512	0.0026	0.0262	26.5712	0.4820	10.9419	11.6494	11.3510	0.0085	0.0025	0.0242
SCEM-FA (r=0.1)	4.2885	3.3627	0.0526	0.0024	0.0295	25.2751	0.4926	11.2260	11.6630	11.1780	0.0088	0.0027	0.0345
SCEM-FA (r=0.3)	29.4503	25.9629	0.0580	0.0085	0.0396	32.0259	0.5540	15.1575	182.9817	122.1159	0.0075	0.0054	0.0478
SCEM-FA (r=0.5)	29.4968	26.4420	0.0573	0.0081	0.0369	32.2464	0.5393	14.8277	180.8541	124.1650	0.0066	0.0053	0.0386
SCEM-FA (r=0.7)	47.2430	40.4975	0.0677	0.0133	0.0486	40.5929	0.7078	23.1352	299.4427	209.2608	0.0053	0.0078	0.0717
SCEM-FA (r=0.9)	47.1152	56.9383	0.0962	0.0130	0.0391	46.5461	0.7981	33.2429	297.6354	208.1662	0.0090	0.0080	0.0967

Table 5. Summary of standard deviations associated with SAC-SMA parameter values obtained from an earlier version of SCEM-FA, for fixed values of r, and also SCEM-UA. Each individual result is based on thirty unique instances for the earlier version of SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

Method	Objective Function Values	
	Average	Standard Deviation
SCEM-UA	13.31669413	0.02331727
SCEM-FA	13.27271167	0.02663205

Table 6. Summary of objective function value lower order statistics for SCEM-FA, relative to SCEM-UA. Each individual result is based on thirty unique instances for SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

Method	Adjustable Model Parameters Average Values												
	UZTWM	UZFWM	UZK	PCTIM	ADIMP	ZPERC	REXP	LZTWM	LZFSM	LZFPM	LZSK	LZPK	PFREE
SCEM-UA	26.8812	29.5574	0.3811	0.0040	0.2722	203.3623	3.7797	250.7199	48.1212	105.5772	0.2364	0.0200	0.1502
SCEM-FA	27.3790	29.1978	0.3919	0.0033	0.2780	216.8556	3.9589	252.2889	50.7311	105.1115	0.2408	0.0212	0.1565

Table 7. Summary of SAC-SMA average parameter values obtained from SCEM-FA and also SCEM-UA. Each individual result is based on thirty unique instances for SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

Method	Adjustable Model Parameters Standard Deviation Values												
	UZTWM	UZFWM	UZK	PCTIM	ADIMP	ZPERC	REXP	LZTWM	LZFSM	LZFPM	LZSK	LZPK	PFREE
SCEM-UA	4.2085	2.6528	0.0512	0.0026	0.0262	26.5712	0.4820	10.9419	11.6494	11.3510	0.0085	0.0025	0.0242
SCEM-FA	3.2647	2.6198	0.0493	0.0020	0.0228	19.6017	0.4135	9.8273	11.2531	10.1219	0.0060	0.0020	0.0198

Table 8. Summary of standard deviations associated with SAC-SMA parameter values obtained from SCEM-FA and also SCEM-UA. Each individual result is based on thirty unique instances for SCEM-FA and also SCEM-UA, in particular, 15,000 (500 for each of the 30 trials) samples generated after convergence to a posterior distribution has been achieved with either the SCEM-FA or SCEM-UA.

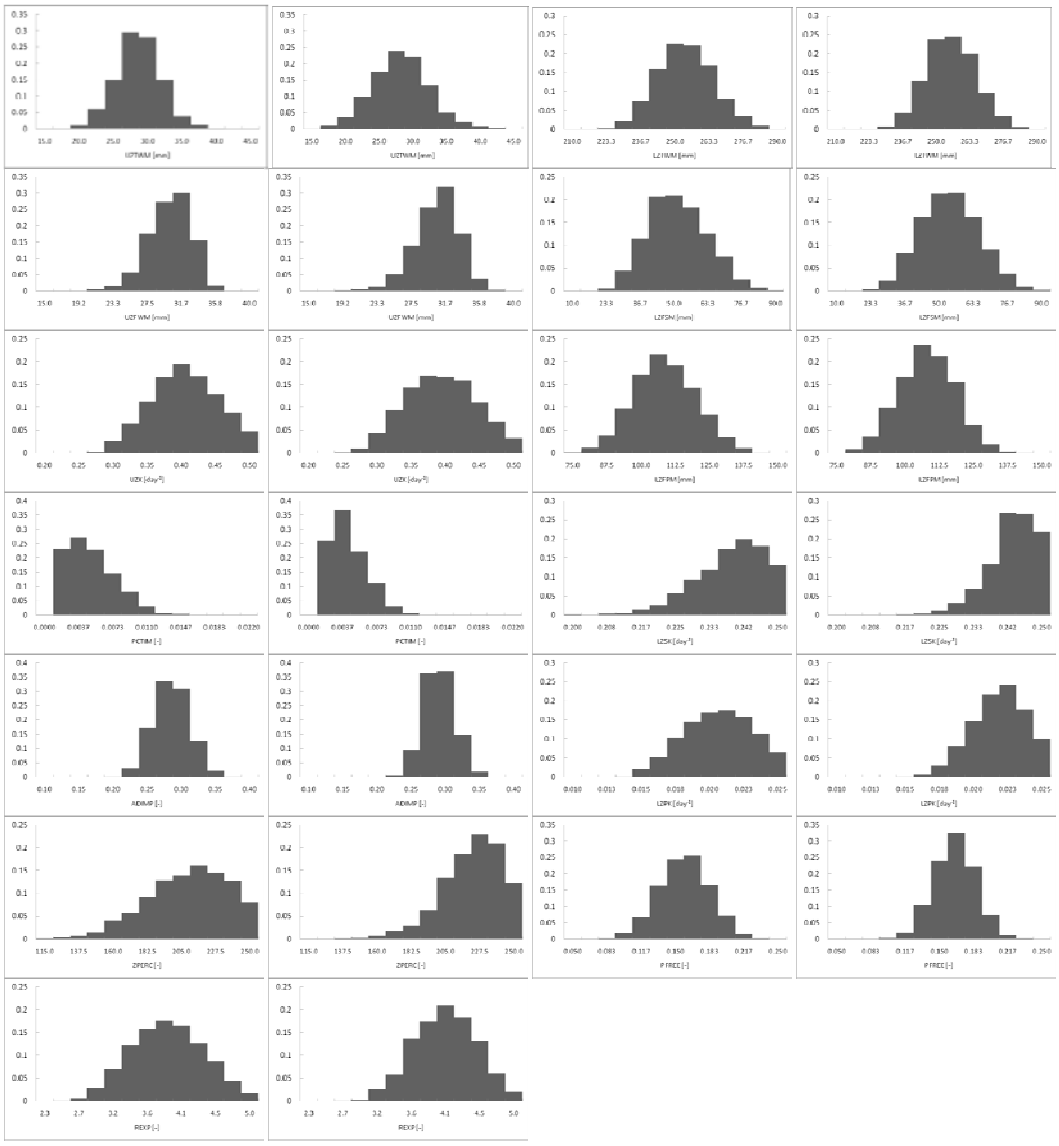


Figure 3. The marginal posterior probability distributions of the SAC-SMA model parameters inferred for the Leaf River watershed using the 15,000 (500 for each of the 30 trials) samples generated with the SCEM-UA (1st and 3rd columns) and SCEM-FA (2nd and 4th columns) algorithms after convergence has been achieved with SCEM-UA and SCEM-FA.

Summary

This report began by outlining the need for hydrologic model calibration and, related, the realistic assessment of hydrologic model uncertainty, which has many benefits, including model comparison and selection, identification of the best water management strategies that reflect the likelihood of outcomes, data collection aimed at improving hydrologic predictions and water management, and regionalization and extrapolation of hydrologic parameters to ungauged basins. Bayesian-based approaches to model calibration, in particular Markov Chain Monte Carlo (MCMC) simulation methods, are a formal means to obtain a realistic and reliable estimate of model uncertainty. However, their application comes at a computational cost. For hydrologic modeling, it was noted that there is a need to design efficient MCMC samplers, and this observation was in fact the motivation for the development of the Shuffled Complex Evolution Metropolis algorithm (SCEM-UA) (Vrugt et al., 2003a). The primary objective of the research and development encapsulated in this report was to improve upon the already existing documented efficiency of the state-of-the-art Bayesian model uncertainty analysis method SCEM-UA (Vrugt et al., 2003a). As with other recent research and development activity that was directed to enhancing the efficiency to the state-of-the-art covariance matrix adaption evolution strategy (CMAES) (Baggett and Skahill, 2010a; Baggett and Skahill, 2010b), the principal approach that was employed to achieve that stated objective was to simultaneously and adaptively construct an approximation to the objective function.

The report followed with some brief background material and then a description of the current methodology that is employed for interfacing a function approximation model with the native SCEM-UA algorithm to improve upon its already existing documented efficiency. Thereafter, based on a comprehensive set of thirty trials using the SAC-SMA soil moisture accounting hydrologic model and local hydrologic data for the Leaf River watershed near Collins, MS, it was clearly demonstrated that SCEM-FA was able to achieve, on average, a 21% reduction in total model calls while inferring the same posterior probability distribution as that obtained with SCEM-UA.

Several opportunities exist for future development (and likely additional efficiency gains) and also application. Numerical experiments are needed to explore how the random number threshold, r, the acceptance rate threshold, a, the parameter, p, an integer value which determines the update frequency for r, and the increment/decrement value embedded in the dynamic adjustment process for r impact overall SCEM-FA performance, relative to SCEM-UA, in terms of efficiency and reliability. A relatively simple and easily implementable approach that would likely yield additional efficiency gains for the current implementation of SCEM-FA would be to monitor the success rate of the function approximation prediction and use that as a basis for deciding whether to perform a forward model call after the function approximation prediction suggests the candidate point is to be accepted. Early SCEM-FA development utilized a k-nearest neighbor cubic radial basis function (RBF) local function approximation model rather than locally weighted projection regression (LWPR). It would be interesting to explore how the two different function approximation models impact overall SCEM-FA performance, relative to SCEM-UA, in terms of efficiency and reliability. Moreover, it could be of potential benefit to explore ways in which the confidence estimate associated with the LWPR function approximation prediction could be beneficially used to improve overall SCEM-FA performance, relative to SCEM-UA, in terms of efficiency and reliability. Additional case studies are needed to further document SCEM-FA performance in terms of efficiency relative to SCEM-UA. And future applications need to focus on model prediction uncertainty. As was mentioned earlier, the methods reported upon in this report should be relatively easy to transfer to other MCMC methods. It is our intent to explore just that with the DREAM MCMC sampler, particularly in light of potential balance issues that have been presented regarding the SCEM-UA algorithm (Vrugt et al., 2008). The code for the SCEM-FA algorithm is available from the first author.

References

Baggett, J., and Skahill, B. (2010a). "Hybrid Optimization using Evolutionary and Approximate Gradient Search for Expensive Functions." (submitted to Engineering Optimization).

Baggett, J. and Skahill, B., 2010b. Hybrid Optimization using Evolutionary and Approx-

imate Gradient Search for Expensive Functions. In: Proc. 2nd Int. Conf. Eng. Opt.

Box, G.E.P., Tiao, G.C. (1973). Bayesian Inference in Statistical Analysis. Addison-Wesley, Reading, MA.

Campbell, E.P., Fox, D.R. and Bates, B.C. (1999). “A Bayesian approach to parameter estimation and pooling in nonlinear flood event models.” Water Resourc. Res. 35 (1) 211-220.

Campbell, E.P. and Bates, B.C. (2001). “Regionalization of rainfall-runoff model parameters using Markov Chain Monte Carlo samples.” Water Resourc. Res. 37 (3), 731-739.

Doherty, J., and Skahill, B.E. (2006). “An Advanced Regularization Methodology for Use in Watershed Model Calibration.” Journal of Hydrology, (327), 564– 577.

Duan, Q.S., Sorooshian, S., Gupta, V.K. (1992). “Effective and efficient global optimization for conceptual rainfall runoff models.” Water Resour. Res. 28 (4), 1015–1031.

Duan, Q., Gupta, V.K., Sorooshian, S. (1993). “A shuffled complex evolution approach for effective and efficient global minimization.” J. Optim. Theory Appl. 76 (3), 501–521.

Gallagher, M., and Doherty, J. (2007). “Parameter estimation and uncertainty analysis for a watershed model.” Environmental Modelling and Software (22), 1000-1020.

Gelman, A., and Rubin, D.B. (1992). “Inference from iterative simulation using multiple sequences.” Stat. Sci., 7, 457-472.

Gupta, H.V., Sorooshian, S., Hogue, T.S., Boyle, D.P. (2003). "Advances in automatic calibration of watershed models.” In: Duan, Q., Gupta, H., Sorooshian, S., Rousseau, A., Turcotte, R. (Eds.), Water Science and Application Series 6, 197–211.

Hansen, N. (2006). The CMA Evolution Strategy: A Comparing Review. In J.A. Lozano, P. Larrañga, I. Inza and E. Bengoetxea (Eds.). Towards a new evolutionary computation. Advances in estimation of distribution algorithms. pp. 75-102, Springer

Harmon, R. and Challenor, P. (1997). “A Markov chain Monte Carlo method for estimation and assimilation into models.” Ecological Modelling, 101, 41-59.

Hastings, W.K. (1970). “Monte Carlo sampling methods using Markov chains and their applications.” Biometrika, 57, 97-109.

Kanso, A., Gromaire, M.-C., Gaume, E., Tassin, B and Ghebbo, G. (2003). “Bayesian approach for the calibration of models: application to an urban stormwater pollution model.” Water Science and Technology 47 (4) 77-84.

Kuczera, G. and Parent, E. (1998). “Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm.” J. Hydrol., 211, 69-85.

Makowski, D., Wallach, D and Tremblay, M. (2002). “Using a Bayesian approach to parameter estimation; comparison of the GLUE and MCMC methods.” Agronomie 22, 191-203.

Matott, LS, Babendreier JE, Purucker ST. (2009). “Evaluating Integrated Environmental Models: A Survey of Concepts and Tools.” Water Resources Research, vol. 45, W06421, doi: 10.1029/2008WR007301

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). “Equations of state calculations by fast computing machines.” J. Chem. Phys., 21, 1087-1091.

Powell, M.J.D. (1992). The theory of radial basis function approximation in 1990, Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions, W. Light Ed., London, U.K., Oxford Univ. Press, 105-210.

Qian, S.S., Stow, C.A. and Borsuk, M.E. (2003). “On Monte Carlo methods for Bayesian inference.” Ecological Modelling 159, 269-277.

Schoups, G., and Vrugt**,** J.A. (2010). “A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic and non-Gaussian errors.” Water Resources Research, doi:10.1029/2009WR008933, In Press.

Schoups, G., Vrugt, J.A., Fenicia, F., and van de Giesen, N.C. (2010). "Inaccurate numerical implementation of conceptual hydrologic models corrupts accuracy and efficiency of MCMC simulation." Water Resources Research, doi:10.1029/2009WR008648, In Press.

Skahill, B., and Baggett, J. (2010). "More efficient Bayesian-based optimization and uncertainty assessment of hydrologic model parameters." (in preparation – 75% complete – to submit to the Journal of Hydrology, Environmental Modelling & Software, or Water Resources Research).

Skahill, B., Baggett, J., Frankenstein, S., and Downer, C.W. (2009). "Efficient Levenberg-Marquardt Method Based Model Independent Calibration." Environmental Modelling & Software (24), 517–529.

Skahill, B., and Doherty, J. (2006). "Efficient accommodation of local minima in watershed model calibration." Journal of Hydrology, (329), 122-139.

Vijayakumar, S., D'Souza, A., and Schaal, S. (2005). "Incremental Online Learning in High Dimensions." Neural Computation, vol. 17, no. 12, pp. 2602-2634.

Vrugt, J.A., Gupta, H.V., Bouten, W. and Soorooshian, S. (2003a). "A shuffled complex evolution metrolopolis algorithm for optimization and uncertainty assessment of hydrologic model parameters." Water Resourc. Res. 39 (8) 1201-1215.

Vrught, J.A., Gupta, H.V., Bastidas, L.A., Bouten, W. and Sorooshian, S. (2003b). "Effective and efficient algorithm for multiobjective optimization of hydrologic models." Water Resourc. Res. 38 (8), 1214-1233.

Vrugt, J.A., ter Braak, C.J.F., Clark, M.P., Hyman, J.M., and Robinson, B.A. (2008). "Treatment of inputuncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation." Water Resources Research, 44, W00B09, doi:10.1029/2007WR006720.